AN ABSTRACT OF THE DISSERTATION OF

Khandakar M. Rashid for the degree of Doctor of Philosophy in Civil Engineering presented on November 19, 2021.

Title: Automated Monitoring of Construction Operations for Data-Driven Decision Making.

Abstract approved:

____

Joseph Louis

The continuous improvement of construction operations requires a systematic approach of monitoring and making appropriate control actions. However, the lack of real-time information hinders this workflow and eventually compromises timely and effective decision-making. Project managers spend a great deal of time and effort to solve problems emerging from lack of timely information, poor coordination, and inaccurate out-of-date data. Emerging technologies like advanced data analytics, the internet of things, and superior computational power can aid in obtaining real-time information and actionable insight from the construction site. This is reflected in the growing use of emerging technologies to automatically monitor construction activities to improve the efficiency of construction management.

The overarching research goal of this dissertation is to advance the body of knowledge and practice by integrating emerging technologies with project monitoring for data-driven decision-making. Specifically, this research develops a systematic framework to automatically identify activities performed by construction resources and then uses this real-time information to optimize the operations for data-driven decision-making. The methods developed in this study are applied to two different types of construction operations: heavy civil construction, and prefabricated construction. For both types of operations, first consumer-grade sensors, such as

inertial measurement units (IMUs), microphones, RFID sensors were used to automatically identify, and track activities performed in the construction site utilizing machine learning and deep learning algorithms. Then the output from the activity identification framework was used as inputs to simulation models for dynamic productivity estimation and optimization of the operation to enable data-driven decision-making. This study contributes to the body of knowledge by providing a means for automated monitoring of construction operations using emerging technologies and assessing the use of simulation modeling for data-driven decision-making.

Automated Monitoring of Construction Operations for Data-Driven Decision Making

by

Khandakar M. Rashid

A DISSERTATION

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Doctor of Philosophy

Presented November 19, 2021
Commencement June 2022

Doctor of Philosophy dissertation of Khandakar M. Rashid presented on November 19, 2021

APPROVED:

Major Professor, representing Civil Engineering

Head of the School of Civil and Construction Engineering

Dean of the Graduate School

I understand that my dissertation will become part of the permanent collection of Oregon State University libraries. My signature below authorizes the release of my dissertation to any reader upon request.

Khandakar M. Rashid, Author

ACKNOWLEDGEMENTS

I would like to express my deepest and heartful gratitude to my advisor Dr. Joseph Louis for his constant support and guidance. I consider it one of the best decisions of my life to join Oregon State University and start my Ph.D. journey with him. His contributions to my success lie beyond my academic realm and extend to my personal struggles as well. I could not thank Dr. Louis enough for making my journey at Oregon State University an enjoyable and memorable one.

I am sincerely grateful to my doctorial committee members Dr. John Gambatese, Dr. Weng-Keen Wong, Dr. Hector Vergara, and Dr. Mariapaola Riggio for making the time to be part of my dissertation committee and for providing me with invaluable feedback about my research. I am also thankful for all my course instructors for being so patient and understanding with their students inside as well as outside of the classroom. I would like to thank our resourceful and amicable staff in the School of Civil and Construction Engineering – especially Cindy Olson, Dana Ainsworth, Kathy Westberg, and Michelle McAllaster for their much-needed support.

My biggest appreciation to my wife, Ananna Ahmed for her relentless support and inspiration. I would like to thank her for her compassion and love in all my ups and downs. Being a researcher herself, she has been a constant source of intellectual inputs throughout my whole academic life. Thank you, dear!

Last but not the least, my heartfelt gratitude to my other family in Bangladesh. First and foremost, to my mother for sacrificing so much in her life in raising me and my brother. At every stage of my life, she always believed in me, and told the cliché out

loud, “I know you can achieve anything you set your mind to, no matter how hard it is”. Thank you for always believing in me. Thank you also to my father and brother for being there for me in all my hardships. I am blessed to have a big family with my aunts, uncles, and all the cousins who are a constant source of wisdom, appreciation, and pleaser in my life. Thank you!

CONTRIBUTION OF AUTHORS

Dr. Joseph Louis assisted with the development of the framework and data collection presented in Chapters 2, 3, 4, and 5.

Colby Swanson assisted with the data collection presented in Chapter 5

TABLE OF CONTENTS

Page

TABLE OF CONTENTS (CONTINUED)

Page

TABLE OF CONTENTS (CONTINUED)

Page

TABLE OF CONTENTS (CONTINUED)

Page

LIST OF FIGURES

LIST OF FIGURES (CONTINUED)

LIST OF FIGURES (CONTINUED)

LIST OF FIGURES (CONTINUED)

LIST OF TABLES

DEDICATION

This dissertation is dedicated to my mom. You taught me how to keep going in life, no matter what happens.

CHAPTER 1: INTRODUCTION

The construction industry is one of the largest in the world worldwide and accounts for about $10 trillion spent on goods and services every year (Barbosa et al. 2017). Despite its strategic importance, construction labor-productivity growth averaged only 1 percent in the construction sector globally, compared to 2.8 percent for the total world economy and 3.6 percent for manufacturing during in last two decades. The reasons behind the productivity stagnation in construction sectors can be broadly categorized into external forces (e.g., increasing project and site complexities, external regulations, uniqueness of product prohibiting mass production, etc.), industry dynamics (e.g., highly fragmented construction, misaligned contractual structures, and incentives, absenteeism at the worksite, etc.), and operational factors (e.g., poor construction methods, inadequate supervision, lack of site control system, insufficiently skilled labor at the supervisory level, lack of communication, non-availability of information, etc.) (Barbosa et al. 2017; Hamza et al. 2019; Shehata and El-Gohary 2011; Teicholz 2013).

Of the above forces, this dissertation focuses on operational factors such as lack of communication between site and supervisor, non-availability of as-built information, lack of site control system. These factors directly impact the ability of project manager/site managers to monitor performance indicators, which reduces their ability to detect and manage the variability and uncertainty of the project activities, thus obstructing timely decision making (Koskela and Howell 2001).

Construction operations require a systematic approach to gaining value and knowledge for continuous improvement. This systematic approach can be described with Deming's cycle, or PDSA (Plan-Do-Study-Act) cycle (Sokovic et al. 2010). An adapted version of Deming's cycle for construction operation is shown in Figure 1.1.

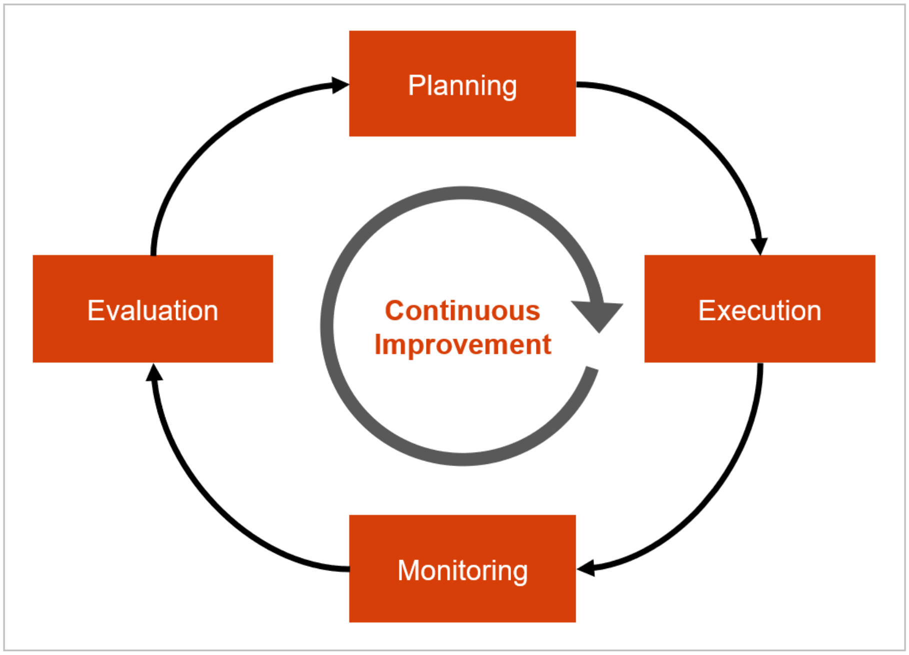


Figure 1.1. Adapted Deming's cycle for construction operation

As seen from Figure 1.1, the cycle for construction operations starts with *planning* the activities, identifying goals, and defining the performance metrics. This is followed by the *execution* of the plans on the actual site. Then starts the *monitoring* of the activities, followed by *evaluation* by comparing the actual and planned performance. Any deviation observed in the monitored operation from the planned operation can prompt the construction manager to take corrective measures and mitigate adverse impacts on project performance. The lean construction management approach also views this as continuous improvement of project workflow (Koskela and Howell 2001).

However, the lack of real-time information hinders this workflow and eventually compromises timely and effective decision-making. Project managers spend a great deal of time and effort to solve problems emerging from lack of timely information, poor coordination, and inaccurate outdated data. Sophisticated management information systems (MIS) or enterprise resource planning (ERP) systems are also limited by the availability of information while assisting the project manager. To get real-time information and actionable insight from the construction site efficiently and reliably, the implementation of emerging technologies such as advanced data analytics, distributed sensor systems, and superior computational power can poetically be helpful. Unfortunately, the construction industry has traditionally been viewed as slow to adopt technological advancement (Goodrum et al. 2011). However, research has shown opportunities as well as evidence of productivity improvement within sectors and processes of the construction industry as a result of utilizing new information technology (Goodrum et al. 2011; Zhai et al. 2009). Thus, it is reflected in the growing use of emerging technologies to automatically monitor construction activities to improve the efficiency of construction management.

1.1. Background

The background for this research will be described in two major categories that encompass the main topics of this study: (1) Automated monitoring of construction operations, and (2) Planning and optimization of construction operations. The first topic focuses on automated data collection methods for real-time information extraction using sensing technologies, such as cameras, location tracking devices (e.g., GPS), radio frequency identification (RFID), inertial measurement units (IMU), etc. and advanced data analytics (e.g., machine learning, deep learning) for progress monitoring. The second category of research deals with simulation (e.g., discrete event simulation) and optimization of construction operations for planning. State-of-the-arts in the abovementioned two categories are briefly discussed below.

1.1.1. Automated monitoring of construction operations

Construction site monitoring and operational analysis is an important contributor to overall project success but has traditionally been a labor-intensive manual process. These manual approaches have been noted to adversely affect the quality of the analysis (Carbonari et al. 2011), minimize opportunities for continuous long-term monitoring (Cheng and Teizer 2013), and result in subjective reports that together hinder proactive and informed decision making. Due to these disadvantages, several research efforts have focused on developing techniques to automatically monitoring activity on the construction site. Researchers in the construction engineering and management domain (CEM) have explored various approaches to recognize, track, and identify activities performed by construction resources. As shown in Figure 1.2, these studies in CEM can be broadly separated into two categories: activity recognition of workers, and activity recognition of equipment.

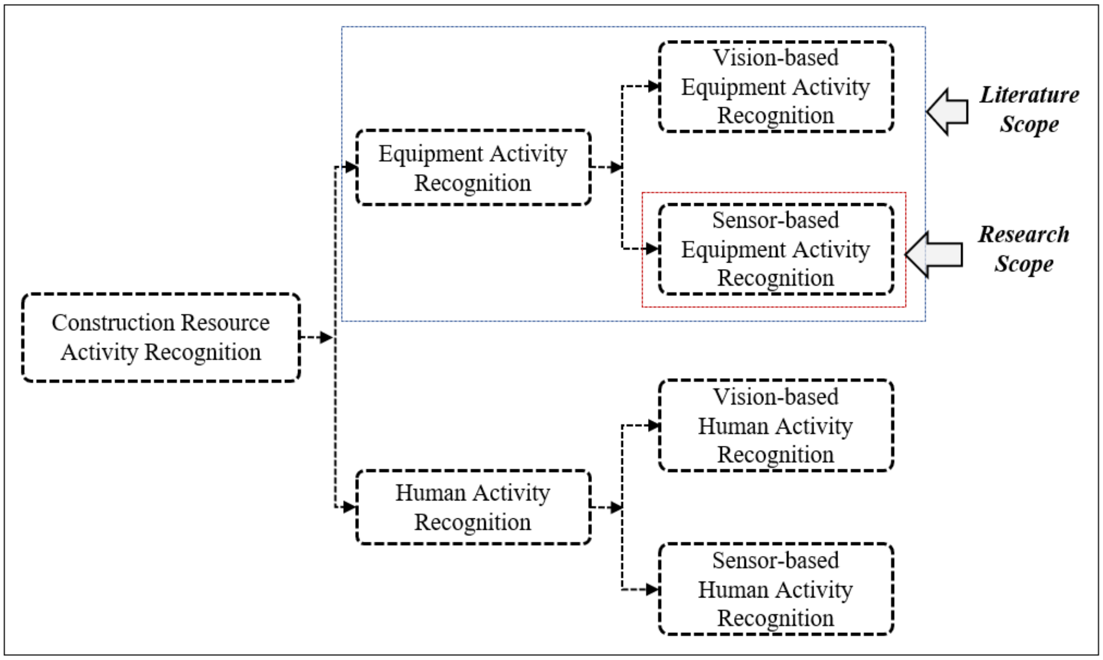


Figure 1.2. Construction resource activity recognition and scope of this study

Based on the primary type of sensor used, these research efforts are broadly categorized into vision-based and sensor-based activity identification. In the latter category, researchers have explored both RTLS (e.g., GPS, UWB) and/or motion sensors (e.g., IMUs) for data collection. After the raw data are obtained, they are processed and analyzed using a variety of analytical methods (e.g., linear classifiers, support vector machines, decision trees, random forests, neural networks, nearest neighbors, etc.) to obtain the desired performance. Table 1.1 provides a summary of representative research efforts towards automatic activity recognition of construction equipment along with the analytical method used.

Table 1.1. Summary of equipment activity recognition research

Techniques	**Reference**	**Objective**	**Analytical Method Used**
Vision-Based Approaches	Kim et al. (2018b)	Interaction analysis between equipment	Spatiotemporal reasoning and image differencing
	Bao et al. (2016)	Operational efficiency analysis	Decision tree
	Golparvar-Fard et al. (2013)	Analysis of construction operations	Support vector machine
	Heydarian et al. (2012)	Analysis of construction operations	Support vector machine
	Rezazadeh Azar and McCabe (2012)	Real-time activity control	Bayesian belief network Hidden Markov Model
	Gong et al. (2011)	Operation analysis and ergonomic studies	Bayesian learning method
	Zou and Kim (2007)	Idle time analysis	Hue, Saturation and Clue Color Space Analysis
RTLS and/or IMU-Based Approaches	Vahdatikhaki and Hammad (2014)	Near real-time simulation	Rule-based approach
	Song and Eldin (2012a)	Look-ahead scheduling	Adaptive modeling
	Akhavian and Behzadan (2015)	Input for simulation modeling	Artificial Neural Network Decision Tree K-nearest neighbor Linear Regression Support Vector machine
	Ahn et al. (2015)	Monitoring operational efficiency	Naïve Bayes, Inquire based learning (IBL), J48, Multilayer perception

	Mathur et al. (2015)	Cycle time measurement	Multi-layer perception Decision Tree Sequential minimal Random Forest Logistic regression Bayes Net Support vector machine

It can be observed from Table 1.1 that previous efforts explored applications of equipment activity recognition for operational efficiency analysis, cycle time measurement, real-time activity control, and modeling dynamic simulation input. Also, various analytical methods such as statistical methods (i.e., Bayesian learning), distance algorithms (i.e., K-nearest neighbor), decision tree, neural networks (i.e., artificial neural network), and rule-based algorithms (i.e., Markov model) have been applied to process collected data. A more detailed examination of these activity recognition efforts is now provided to set the context for this research.

1.1.1.1. Vision-based activity identification of equipment

Many previous efforts have adopted vision-based techniques to identify the activities of construction equipment. Zou and Kim (2007) used image processing to quantify the idle times of hydraulic excavators by identifying only two states of the excavator: idle and busy. Azar and McCabe (2012) proposed an activity identification framework using rational events to recognize the dirt-loading activities of an excavator. Bao et al. (2016) investigated the use of long-sequence videos to automatically detect, track, and identify activities of an excavator and a dump truck. In a similar effort, excavators and dump trucks were also used to measure the performance of earthmoving operations utilizing image frame sequences (Kim et al. 2018b). The concept of the bag-of-video-feature-words model was extended using unsupervised classifiers into the construction domain to learn and classify labor and equipment activities (Gong et al. 2011).

Vision-based techniques have shown promising results in tracking construction resources and identifying their operations. However, these techniques provide very limited information based on the field of view of the cameras used. It is challenging to maintain a direct line of sight to targeted resources due to a high level of noise (e.g., entity overlap, moving backgrounds, varying light conditions, etc.) on dynamic construction sites. These challenges can be overcome by adopting motion sensors that are not thus constrained.

1.1.1.2. Sensor-based activity identification of equipment

As opposed to vision-based methods, sensor-based approaches for activity identification uses different sensors (e.g., GPS, inertial measurement unit, etc.) to capture the location and motion of the equipment and then extract activity information from them.

Location-based activity identification: Vahdatikhaki and Hammad (2014) proposed a multi-step data processing framework combining location and motion data to improve the accuracy of the localization to enhance the performance of equipment state-identification. Song and Eldin (2012) developed an adaptive real-time tracking of equipment operation based on their location to improve the accuracy of project look-ahead scheduling. Su and Liu (2007) presented a framework that used dynamic geometric data of resources and extracted construction operational data from them. In another effort, Wang et al. (2012) proposed an automated methodology for tracking earthmoving operations in near real-time by attaching low-cost RFID tags to hauling units (trucks) and attaching fixed RFID readers to designated gates of projects' dump areas. Remote tracking technology was also used to develop 3D animation of the equipment, and extracting equipment operations from the animation (Akhavian and Behzadan 2012a). Teizer et al. (2018) explored the feasibility of ultra-wideband (UWB) technology for real-time tracking and monitoring of resources.

IMU-based activity identification: Although location-based operation tracking can identify the state and operation of construction equipment at a coarse level (e.g.: *idle* and *busy* states), it is incapable of classifying the activities performed by equipment when it is stationary. Such limitations of location-based operation tracking inspired researchers to explore the feasibility of both independent (Ahn et al. 2015) and smartphone embedded (Akhavian and Behzadan 2015; Mathur et al. 2015) inertial measurement units (IMUs) for automated equipment activity identification. Ahn et al. (2015) used a low-cost accelerometer mounted inside the cabin of an excavator to collect operational data of an earthmoving worksite. Several classifiers were tested to classify three different states (i.e. engine-off, idle, and busy) of an excavator. Mathur et al. (2015) utilized a smartphone-embedded accelerometer by mounting it inside an excavator cabin to measure various activity modes (e.g. wheel-base motion, cabin rotation, and arm movement) as well as duty cycles. Akhavian and Behzadan (2015) adopted a similar approach by attaching a smartphone to the cabin of a front-end loader to collect accelerometer and gyroscope data during an earthmoving operation, upon which several machine learning algorithms (i.e., ANN, DT, KNN, LR, SVM) were tested. Their study also investigated the impact of different technical parameters such as level of details, and selection of features on the performance of different classification algorithms. The same approach and technical parameters were further extended for construction workers (Akhavian et al. 2015).

1.1.2. Planning and Optimization of Construction Operations

In the context of this research, simulation modeling is the process of creating and analyzing a virtual model of a real-world process to predict and forecast its performance. Simulation modeling has been widely explored for on-site and off-site construction processes in several previous studies (Afifi et al. 2017; Akhavian and Behzadan 2013; AlDurgham and Barghash 2008; Altaf et al. 2015b, 2018; Hammad et al. 2002; Jeong et al. 2011; Louis et al. 2014; Louis and Dunston 2016a; Zhang 2004). A detailed review of the studies aiming to optimize construction operations is provided below.

1.1.2.1. Simulation modeling in on-site construction

There has been a significant body of work on the application of different simulation techniques (e.g., discrete-event simulation, agent-based simulation, etc.) to improve the efficiency of on-site construction operations. Louis and Dunston (2016) This study provides a framework for real-time monitoring of earthmoving operations using sensor data and finite-state machines. The authors also demonstrate the utility of discrete-event simulation modeling in both the planning and construction phase by advancing the model with construction resources. Akhavian and Behzadan (2013) developed a methodology to generate operational knowledge from multimodal data collected from various sensors (e.g., load cell, ultra-wideband, etc.) attached to earthmoving equipment. The generated knowledge was further utilized to develop an accurate and realistic simulation model. The methodology was validated using laboratory-scale experiments. Lu and Olofsson (2014) proposed a framework consisting of building information modeling (BIM) and DES to enable the integration of DES in the planning and follow-up of construction activities. Vahdatikhaki et al. (2013) proposed a framework integrating tracking technologies used in Automated Machine Guidance (AMG) with simulation-driven 4D modeling to use the simulation tool as a proactive monitoring and planning tool in earthmoving projects. A laboratory-scale experiment was conducted to validate the proposed framework. Shitole et al. (2019) integrated a discrete-event simulation model with reinforcement learning and neural network to optimize earthmoving operations. The near-optimum policies generated from the proposed framework require minimum human guidance and outperformed human-design heuristics. Kim and Kim (2010) aimed to develop a multi-agent-based simulation model to analyze the traffic flow of construction equipment on construction sites. DES modeling cannot simulate the continuous dynamic behavior of the construction equipment, and thus this study intends to address this limitation by describing the behavioral characteristics of construction equipment using a multi-agent-based simulation system.

The application of simulation tools in planning and managing underground infrastructure constructions was investigated (Ruwanpura and Ariaratnam 2007). This study shows the usefulness of simulation and analytical tools in capturing the risks and uncertainties of underground infrastructure construction projects and assisting in decision-making. González and Echaveguren (2012) proposed a dynamic modeling framework based on DES by integrating traffic models and sustainable goals in road construction operations. A hypothetical project was studied to validate the framework and the results demonstrated that the proposed framework could optimize the number of trucks and front-end loaders to minimize the emission level. Chen et al. (2012) presented an intelligent scheduling system based on simulation modeling and integrating the major construction factors such as schedule, cost, space, manpower, equipment, and material simultaneously in a unified environment. Moreover, the evolutionary algorithm was utilized to achieve near-optimum distribution of manpower, equipment, material, and space according to project objectives and constraints. The case studies demonstrated higher effectiveness of the proposed framework compared to traditionally used scheduling tools such as Primavera and MS Projects.

Zhang et al. (2014) proposed a DES model to estimate the emissions and noise generated from construction equipment from earthmoving projects. The case study demonstrated that the proposed framework is more convenient and accurate in accounting uncertainties, randomness, and dynamics in quantifying the emissions and noise compared to field measurement. Chan and Lu (2008) presented a DES modeling approach to improve the effectiveness of the material handling system in a precast viaduct construction project. The knowledge generated from the simulation model was added to the experience of the site manager and project director to assist them in designing the material handling system.

Another study used DES modeling for productivity estimation of a sanitary trunk tunnel project (Chung et al. 2006). The Bayesian updating method was used to update the simulation model with the most recent field data, and the result demonstrated improvement to the planning predictions compared to the initial estimates. In another effort, research presented a BIM-integrated simulation framework incorporating critical factors affecting productivity at the operation level to predict productivity dynamics at the construction planning phase (Jeong et al. 2016). The integration of BIM with the construction operation simulation enabled the adoption of construction plans according to the project changes. The findings demonstrated that the framework was reliable in predicting productivity dynamics. Mao and Zhang (2008) developed a framework for construction process reengineering by integrating lean principles and computer simulation techniques. Instead of labeling the activities as value-adding and non-value-adding, this study classifies them into main and supportive activities which is more effective in modeling the construction workflow. The results showed the effectiveness of the developed framework in assessing the efficiency of the re-engineered construction process.

1.1.2.2. Simulation modeling in off-site construction

Significant research has been conducted in optimizing the construction process in off-site constructions. Altaf et al. (2018) proposed an integrated production planning and control system for panelized home building using DES and radio frequency identification (RFID)- based tracking. A discrete and continuous simulation approach was also explored to optimize the production of off-site construction elements (Afifi et al. 2017). AlDurgham and Barghash (2008) proposed a simulation-based approach to facilitate decision-making for planning layout, material handling, scheduling, and manufacturing processes and resources for off-site house building. Ng et al. (2009)

used an activity-based construction simulation tool to quantify the productivity of resources for a residential prefabricated construction project. The results showed that the developed tool could successfully establish the most suitable strategy to improve the logistics of material handling. Wang and Abourizk (2009) developed a special modeling system to build large-scale simulation models for industrial constructions. The proposed system can simulate activities, product flows as well as information flows. Several simulation experiments were conducted to test the effectiveness of the proposed system.

Liu et al. (2015) investigated the potential of an automated planning tool to improve productivity and balance the production line in a panelized construction factory by integrating BIM and DES. A case study was conducted of a production line for light gauge steel panels and the results showed that the simulation modeling was useful for planning and improved production performance. Goh and Goh (2019) conducted a simulation study for prefabricated volumetric unit construction by utilizing lean concepts to evaluate the application of simulation to improve modular construction efficiency. The baseline simulation model was developed and key lean principles, such as total quality control management, E-Kanban based Just-In-Time deliveries. The results demonstrated that the application of simulation and lean principles reduced cycle time and process time and increased process efficiency and labor productivity. Cheng et al. (2020) developed a system dynamics model to simulate the impacts of governments incentive strategies on the prefabricated construction industry by using an evolutionary game process between the government and the contractor. Du et al. (2019) investigated the potential of a multi-agent-based model to simulate the coordination mechanism of the management strategies for design changes in prefabricated construction. The proposed simulation was able to test different design changes and their effectiveness in the managerial decision-making process.

Murray et al. (2003) developed a virtual environment that can be used to interactively design prefabricated buildings and view a real-time simulation of the construction process. This aims to present the dynamics of the future-home construction site by enhancing the traditional drawing and Gantt charts. Altaf et al. (2015) proposed a simulation-based monitoring platform for the prefabricated construction process by integrating DES and an RFID system. The RFID was used to capture real-time production states and those were fed into the simulation model for real-time simulation results. Neill et al. (2020) proposed a virtual model-based simulation system to automatically model prefabricated building components using data available in BIM models. This system improves the efficiency of the on-site assembly analysis by optimizing the process for contractors.

1.2. Research Gaps and Point of Departure

Even though previous research has explored the possibilities of real-time, automated data collection, remote monitoring, simulation, and optimization of construction operations there is a lack of in-depth work in synthesizing different techniques and demonstrate the feasibility of such system to be applied in real-world construction projects. Moreover, the 4^{th} industrial revolution requires an amalgamation of modern sensing technologies, advanced data analytics, engineering visualization, simulation, and optimization to run ever-so-complicated construction projects effectively and successfully (Alaloul et al. 2020). Based on the review of literature conducted, the following gaps in the body of knowledge are identified:

I. There is a lack of knowledge on the applicability of consumer-grade sensors in monitoring construction activities in real-time.

II. A complete methodological framework optimizing physical (e.g., sensor placements, number of sensors, etc.) and analytical parameters (e.g., type of algorithms, resolution of activities, etc.) to monitor construction activities is missing from the literature.

III. There is a lack of knowledge on how to use real-time activity information to facilitate the planning and optimization of construction operations.

Towards this end, research is required to provide better means of monitoring the activities for data-driven decision-making.

1.3. Research Goal and Objectives

To fulfill the abovementioned knowledge gap, the overarching goal of this dissertation is **to integrate emerging technologies with project monitoring for data-driven decision-making for construction operations.** Two major objectives are identified to achieve this goal:

Obj 1. Automatically identify states (i.e., location and activity) of the resources in the construction site.

Obj 2. Develop dynamic simulation tools to analyze and optimize the operations.

The specific research questions that this study aims to answer are:

Q1. What type of consumer-grade sensors can be used on construction sites to ensure a safe and reliable source of data collection?

Q2. What type of data analytics techniques is appropriate to automatically identify activities from the resources?

Q3. What are the key considerations that ensure optimum performance of the activity identification framework?

Q4. Can simulation tools be used to provide dynamic predictions using the outcomes of the activity identification framework?

Q5. What type of feedback can be provided from the monitoring and simulation for timely and effective decision-making?

1.4. Scope of the Dissertation

The scope of this study entails two different types of construction operations, and they are heavy civil construction and prefabrication construction.

1.4.1. Heavy civil construction

One of the main activities associated with heavy civil construction is earthmoving which encompasses the moving, removing, and/or adding soil or rock as a part of engineering work. Some of the most common projects that require earthmoving are the construction of roads, railway beds, dams, canals as well as commercial and residential buildings. These types of operation are primarily equipment-driven and major heavy equipment used in the earthmoving operation are excavators, backhoe loaders, front-end loaders, bulldozers, graders, scrapers, compactors, and dump trucks. Earthmoving operations are mostly cyclical, and a major part of the activity involves digging loading soil to dump trucks using excavators, and loaders, transporting the soil, and dumping in the dump zone. To successfully manage earthmoving operations, it is essential to monitor the activities of the heavy equipment. Moreover, the information for the cyclical operation can be collected, analyzed, and used to plan before construction begins and to optimize during the construction enabling data-driven decision making.

1.4.2. Prefabricated construction

Prefabricated or off-site construction is the process of building various components and/or modular units of the structure at a manufacturing site and transporting those to the job site and assembling them. Prefabricated construction is becoming popular due to its documented advantages over traditional stick-built methods in terms of reduced waste and construction time, more control over resources and environment, and easier implementation of novel techniques and technologies in controlled practical settings.

The highest level of modularization is the volumetric construction where modular units in the form of a three-dimensional unit for building are constructed off-site and minimal work is left to assemble it on site. A wide variety of built products are constructed in modular construction factories ranging from single-family homes to multi-family and office buildings. The construction operation inside a modular construction factory is cyclical where the modular units move from one workstation to another and various panelized components of the buildings (e.g., walls, ceiling, etc.) are added and assembled. Thus, to efficiently manage the operation inside a modular construction factory, it is important to monitor the activities that are happening in different workstations. These activities can be tracked and analyzed for better planning and decision-making during the construction process.

1.5. Outline of Dissertation

To achieve the goal and answer the questions, this research study is designed to follow the research flow shown in

Figure 1.3. This dissertation consists of six (6) chapters.

1. Chapter 1 provides an overview and introduction of the study. This chapter provides insights regarding the background of the study, the motivation behind the research, and an outline of the dissertation.   Chapter 1 also includes a brief literature review covering the application of emerging technology for automated data collection from the construction site.

2. Chapter 2 titled "Automated Activity Identification of Heavy Civil Equipment" presents a framework to automatically identify activities of construction equipment from heavy civil construction sites. This portion of the study focuses on developing a model with the application of consumer-grade sensors and machine learning and deep learning techniques to identify and track activities performed by equipment in the earthmoving site. In doing so, important physical and technical parameters were investigated to ensure the optimum performance of the model. Two papers describing this topic have been published in the Elsevier Journal of Advanced Engineering Informatics, and Frontiers in Built Environment (Rashid and Louis 2019b, 2020a). Chapter 2 is adopted from Manuscript #1 and Manuscript #2.

3. Chapter 3 titled “Dynamic Prediction of Earthmoving Operation” presents a case study showing how simulation modeling can be used to provide dynamic predictions for earthmoving operations. A real-world earthmoving site was used for the case study. The main inputs for the simulation modeling were obtained from the output of Chapter 2. A paper describing the content of this chapter will be submitted to a conference. This chapter represents Manuscript #3.

4. Chapter 4 titled “Automated Activity Identification in Modular Construction Factories” presents a developed framework to automatically identify different activities inside a modular construction factory using multiple sources of data. A manuscript describing the content of this chapter has been published in the Elsevier Automation in Construction journal (Rashid and Louis 2020b) and another one has been accepted in the IEEE Winter Simulation Conference 2021. Chapter 4 is a modified version of Manuscript #4 and Manuscript #5.

5. Chapter 5 titled “Optimizing Labor Allocation in Modular Construction Factory” presents a case study using simulation modeling and evolutionary algorithm to optimize the allocation of labor working at different workstations inside a modular construction factory. The content of this chapter has been published in the proceedings of Winter Simulation Conference, 2020 (Rashid and Louis 2020c).

6. Chapter 6 comprises the conclusions section of this study. This chapter includes a summary of how the research objectives were met, major findings, key contributions to the body of knowledge and practice, as well as study limitations and potential future research directions.

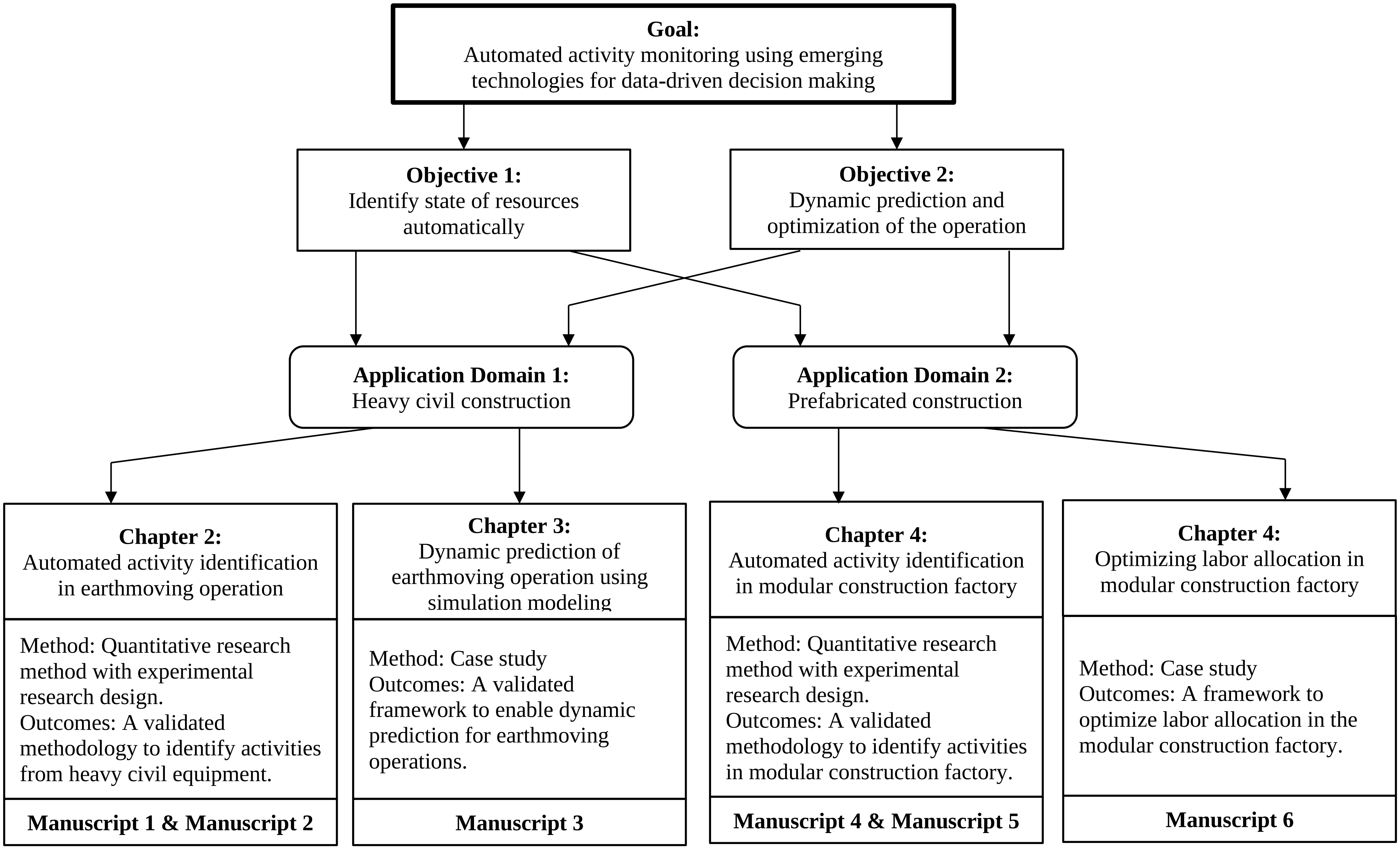


Figure 1.3 Research flow

CHAPTER 2: AUTOMATED ACTIVITY IDENTIFICATION OF HEAVY CIVIL EQUIPMENT

The content of Chapter 2 is an adapted version of the following manuscripts:

1. Rashid, K., Louis, J., (2019). "Times-series data augmentation and deep learning for construction equipment activity recognition." Advanced Engineering Informatics, 42, 100944. DOI: https://doi.org/10.1016/j.aei.2019.100944

2. Rashid, K., Louis, J., (2020), "Automated activity identification for construction equipment using motion data from articulated members." Frontiers in Built Environment. 5. DOI: https://doi.org/10.3389/fbuil.2019.00144

2.1. Introduction

Real-time monitoring and assessment of construction resources in a heavy civil project have always been a challenge due to the unique, dynamic, and complex nature of each construction site and operation. The ability to automatically classify activities performed by various equipment in real time can aid in making timely tactical operational decisions that can lead to increased fleet productivity, reduced operation time and cost, and minimized idle times. Such endeavors that require the identification of individual sequential tasks of equipment (e.g., excavator loading, swinging full, truck dumping) have traditionally been performed manually through human observation, making it extremely labor and time intensive. Meanwhile, the development of low-cost micro-electro-mechanical systems (MEMS) with rapidly evolving computing, networking, and storage capabilities present new opportunities in the real-time equipment activity recognition domain. These sensors, especially inertial measurement units (IMU) are already commercially available on modern heavy equipment and are used presently on the field to determine the position of the cutting edge for automated machine guidance and control but have yet to be utilized for measuring activity cycle-times and equipment productivity.

Previous efforts have explored the use of IMU and machine learning algorithms to identify activities of heavy equipment. However, no methodological framework is present in the literature demonstrating the appropriate use of IMU sensors for activity identification. Moreover, previous studies used different classification algorithms that use time-series sensor data from accelerometers and gyroscopes. These studies utilized pattern recognition approaches such as statistical models (e.g., hidden-Markov models); shallow neural networks (e.g., Artificial Neural Networks); and distance algorithms (e.g., K-nearest neighbor) to model and analyzes the time-series data collected from sensors mounted on equipment. These methods necessitate the segmentation of continuous operational data with fixed or dynamic windows to extract statistical features. This heuristic and manual feature extraction process is limited by human domain knowledge and only can learn using human-specified shallow features. However, recent developments in deep neural networks, specifically recurrent neural networks (RNN), presents new opportunities to model sequential time-series data with recurrent lateral connections. RNN can automatically learn high-level representative features through the network, instead of being manually designed, making it more suitable for complex activity recognition. However, the application of RNN requires a large training dataset which poses a practical challenge to obtain from construction equipment in the real world.

This study presents a data-augmentation framework for generating synthetic time-series training data for an RNN-based deep learning network to identify equipment activities accurately and reliably. Three different equipment in heavy civil, excavator, loader and dump truck were used to validate the framework. Data were collected from three actual earthmoving projects. The deep learning activity identification framework presented in this study outperforms the traditionally used machine learning classification algorithms for activity recognition regarding model accuracy and generalization. Moreover, this study demonstrates the optimum number and placement of the sensors to achieve maximum performance of the model.

2.2. Background

The construction industry is recognized to have lower productivity when compared to other industries that produce engineered products like manufacturing (Cheng et al. 2017). One major contributing factor is the temporary and transient nature of most operations, which make it difficult to implement systems that collect and analyze data to provide insights into that operation. Thus, one of the steps towards productivity improvement is to improve the techniques of assessing and monitoring the performance of key resources. Owning and maintaining heavy construction equipment contributes to a large portion of the total project costs, especially in the case of heavy civil projects. Therefore, identifying and tracking their activities plays an important role in measuring their performance, which itself is the primary prerequisite to enable improving performance. Automated, real-time, and reliable activity recognition of heavy construction equipment is thus a necessary step that enables several other practical applications such as automated cycle-time analysis

(Kim et al. 2018a; Mathur et al. 2015; Vahdatikhaki and Hammad 2014), productivity monitoring (Gong and Caldas 2009; Ok and Sinha 2006; Zou and Kim 2007), safety applications (Cheng and Teizer 2013; Rashid et al. 2017; Rashid and Behzadan 2018; Seo et al. 2015), an environmental assessment (Ahn et al. 2015; Martín-Garín et al. 2018), near real-time simulation inputs (Akhavian and Behzadan 2015; Louis and Dunston 2016b; Vahdatikhaki and Hammad 2014), and applications in AR/VR visualization (Behzadan and Kamat 2012; Dong and Kamat 2013; Louis and Dunston 2016a). To develop an effective activity recognition framework for construction equipment, several previous studies have explored the feasibility of using location data and/or time-series vibration data from inertial measurement units (IMU) (Ahn et al. 2015; Akhavian and Behzadan 2015; Mathur et al. 2015).

2.2.1. RTLS and IMU-based construction equipment activity recognition[1]

2.2.2. Deep learning for sensor-based activity recognition

Extensive research studies have been conducted that implement deep learning algorithms to develop activity recognition frameworks, especially in human activity recognition (HAR). Some of the most common types of deep learning models used in activity recognition tasks are deep neural network (DNN), convolutional neural network (CNN), recurrent neural network (RNN), deep belief network (DBN), and stacked autoencoder (SAE). Vepakomma et al. (Vepakomma et al. 2015) used a DNN model to identify indoor activities of elderly people. In doing so, hand-engineered features were extracted from the wrist-worn sensors, and then those features were fed into a DNN. In another similar effort, Walse and Dhakaskar (Walse and Dharaskar

[1] Please refer to Section 1.4.1 for the literature review of this part.

2016) performed principal component analysis (PCA) before feeding the features to the DNN model. In these efforts, authors only used DNN as a classification model after hand-crafted feature extraction, which may not generalize the model optimally, and may cause a shallow network. To improve the performance of DNN, researchers used a higher number of hidden layers to automatically extract features and generalize the deep network (Hammerla et al. 2016). Improved performance of automated feature extraction using more hidden layers indicates that when HAR data are multi-dimensional and activities are more complex, more hidden layers can help train the deep network by strengthening their representation capability (Bengio 2013).

Convolutional neural network (CNN) is another deep learning model which is capable of automatic extraction of features from signals and it has achieved promising results in the HAR domain. In earlier work, each dimension of the sensor was treated as one channel (like RGB in an image), and then the convolution and pooling were performed separately (Zeng et al. 2014). In another study, a CNN framework was proposed to automate feature learning from the raw input to unify and share weights in multi-sensor data using 1D convolution (Yang et al. 2015). This approach allowed higher-level abstract representation of low-level time-series signals. In another similar work, passive RFID data were directly fed into deep CNN for activity recognition instead of selecting features and using a cascade structure that first detects objects from RFID data followed by predicting the activity.

These de-facto standard approaches of activity recognition treat individual dimensions of the sensor data statistically independently. Thus, each dimension of the data is converted into feature vectors without due consideration of their broader temporal context. To address this, recurrent neural network (RNN) incorporates temporal dependencies of sensor data streams, which is more appropriate for activity recognition than considering the data stream independently. Long-short term memory (LSTM) cells are often incorporated with RNN, serving as the memory units through the gradient descent steps of the RNN. Inoue et al. (Inoue et al. 2016) proposed a deep RNN-based activity recognition framework from raw accelerometer data and investigated various architectures of the model to find the best parameter values. Ordóñez and Roggen (Ordóñez and Roggen 2016) developed LSTM based RNN for multimodal wearable activity recognition which can perform sensor fusion naturally, does not require expert knowledge in designing features, and explicitly models the temporal dynamics of the feature vectors. This framework outperformed the previous results by up to 9%. Hammerla et al. (Hammerla et al. 2016) explored the RNN approach for wearable activity recognition by introducing a novel regularization approach and illustrated that the developed model outperformed the state-of-the-art non-recurrent approaches on a large benchmark dataset.

RNN, specifically LSTM networks, have the capability of modeling sequential time-series data by automatically extracting high-level representative features, and considering the temporal relationship among each time step of the sensor data. This holds a lot of promise for its application in construction equipment activity recognition. However, training an LSTM network requires a large training dataset which is a practical challenge to obtain from the construction equipment in the real world. This limitation is addressed by generating a large volume of synthetic training data from a smaller amount of collected data using data augmentation techniques that will be reviewed in the next section.

2.2.3. Data augmentation in shallow/deep learning

Data augmentation is a technique that enhances a limited amount of data by transforming the existing samples to create new data (Um et al. 2017). This technique has been implemented to generate synthetic training data in the computer vision (Charalambous and Bharath 2016; D'Innocente et al. 2017; He et al. 2016; Liang and Hu 2015; Radford et al. 2015), speech recognition (Jaitly and Hinton 2013; Naoyuki Kanda 2013; Schlüter and Grill 2013), and time-series classification (Forestier et al. 2017; Le Guennec et al. 2016a; Um et al. 2017) domains. Charalambous and Bharath (Charalambous and Bharath 2016) introduced a simulation-based methodology that can be used for generating synthetic video data and sequence for machine/deep learning gait recognition algorithms. D'Innocente et al. (D'Innocente et al. 2017) proposed an image data augmentation technique that zooms on the object of interest in an image and simulates the object detection outcome of a robot vision system to bridge the gap between computer and robot vision.

Most advanced object recognition algorithms utilize various image augmentation techniques, such as flipping, rotating, scaling, cropping, translating, and adding Gaussian noise to generate synthetic data for training and testing machine/deep learning algorithms (Ding et al. 2016; Liang and Hu 2015; Radford et al. 2015). In the speech recognition domain, studies have investigated vocal tract length normalization (Jaitly and Hinton 2013; Naoyuki Kanda 2013), speech rate, and frequency-axis random distortion (Naoyuki Kanda 2013), label-preserving audio transformation (Schlüter and Grill 2013) and evaluated those methods improve learning algorithms.

Despite the frequent implementation of data augmentation techniques in the computer vision and speech recognition domain, data augmentation in time series classification has not been deeply investigated yet (Um et al. 2017). Guennec et al. (Le Guennec et al. 2016b) proposed two-time series data augmentation techniques; window slicing, and time-warping to train a convolutional neural network (CNN). Forestier et al. (Forestier et al. 2017) introduced dynamic time warping (DTW) for time series classification to reduce the variance of a classifier. Um et al. (Um et al. 2017) proposed the most comprehensive set of time series data augmentation techniques to monitor the Parkinson patient using wearable sensors. This research will add to the body of knowledge on time-series data augmentation by applying more transformations and by using it to enable activity recognition for construction equipment.

2.3. Research gaps and point of departure

The following research gaps were identified from the review of current work from the domains of construction equipment activity identification, deep learning application, and time-series data augmentation.

(1) <u>Lack of features with long-term temporal dependency in equipment activity identification:</u> Existing methods that apply pattern recognition approaches towards identifying equipment activity from time-series sensor data require manual extraction of statistical features, which is a time-consuming process, and limited by the need for domain-specific knowledge. Moreover, these

approaches do not consider the long-term temporal dependency between time steps of the time-series data. These limitations could be eliminated by automatically extracting highly representative features, containing the temporal dynamics of the sensor data using deep learning models.

(2) Lack of application of deep learning in construction: Deep learning techniques have hitherto not been used in the construction domain due to the practical challenges posed by the requirement for a large amount of training data. This research will address this limitation by synthesizing data using augmentation techniques.

(3) Lack of large amount of reliable field data: Application of deep learning techniques requires large amount of training data. Time-series data augmentation techniques are a relatively new area of research that can potentially reduce the manual field data collection effort significantly. This chapter will add to the literature in this domain by utilizing more transformations and by providing a practical application of its techniques.

By considering the above specific gaps in the literature, the overall goal for this chapter is framed as enabling activity identification for construction equipment using an LSTM network trained with synthetic data. This research goal is accomplished in this chapter through the pursuit of these specific research objectives:

(1) Develop a deep learning activity recognition framework for construction equipment using motion data of articulated elements (e.g., bucket, boom, arm, etc.) of the equipment.

(2) Develop time-series data augmentation techniques to generate synthetic training datasets to train the LSTM network with a large volume of training for better performance.

(3) Evaluate the performance of the LSTM network by comparing it with traditional shallow networks (e.g., artificial neural network (ANN)) and determine the impact of data augmentation techniques on the performance of the training models.

(4) Determine the optimum location on equipment for sensor placement.

2.4. Methodology

Figure 2.1 illustrates the overall methodology used in this study to achieve the overarching research goal.

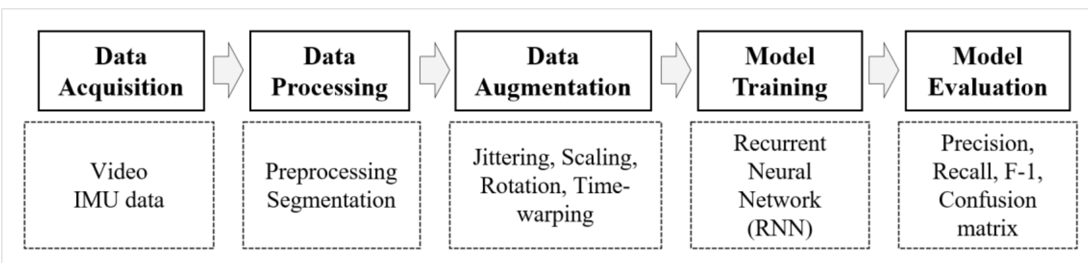


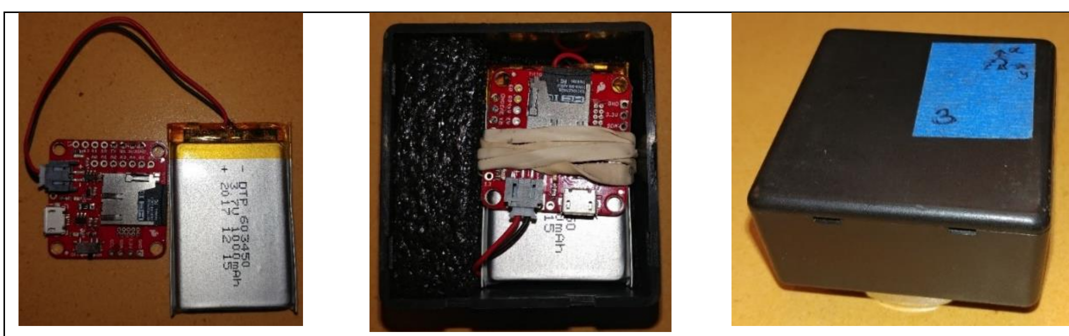
Figure 2.1 Overall methodological architecture of this study

The data acquisition process consists of capturing two different types of sensor data: IMU sensor data for analysis, and video data for validating the methodology. In the data processing phase, the IMU data were first preprocessed by filtering out hardware noise and then eliminating other inconsistencies in collected data. The video was used as a reference for labeling the time-stamped data, while the sensor data were used for data augmentation to generate synthetic data. Four different data augmentation techniques (e.g., *Jittering, Scaling, Rotation*, and *Time-warping*) are implemented to generate synthetic training data. The augmented training data are used to train the RNN (LSTM network). Finally, the model is evaluated using precision, recall, F-1 score, as well as a confusion matrix. Each of the steps shown in Figure 2.1 is explained in the following subsections.

2.4.1. Data acquisition

This study uses multiple inertial measurement units (IMU) mounted with a 3-axis accelerometer and a 3-axis gyroscope. The primary reason for using multiple IMUs is to explore the feasibility of utilizing the motion of various articulated implements of the equipment to identify its activities. Moreover, equipment manufacturers and third-party companies have started using motion sensors in their equipment to locate the cutting edge for automated machine guidance (AMG). The use of multiple IMUs in articulated implements in this chapter mimics the placement of those sensors so that the developed framework can be extended towards activity recognition.

The IMUs used in this research are equipped with a microSD card for logging data and require a 3.7-volt 1000 mAmp battery as shown in Figure 2.2.



IMU with battery and microSD card	IMU system placed on a magnetic box	Magnetic box with number and axes tag

Figure 2.2. Setup of the IMU and the protective box

The IMUs are placed in a robust plastic box stuffed with Styrofoam material to prevent movement and to withstand vibration and any impact from the debris. A powerful magnet at the bottom of each box made sure the reliable placement of the IMUs on the metallic surface of the equipment. Figure 2.3 shows the different locations of the equipment where IMUs are attached. Three IMUs are used for excavator and front-end loader, and one for dump truck.

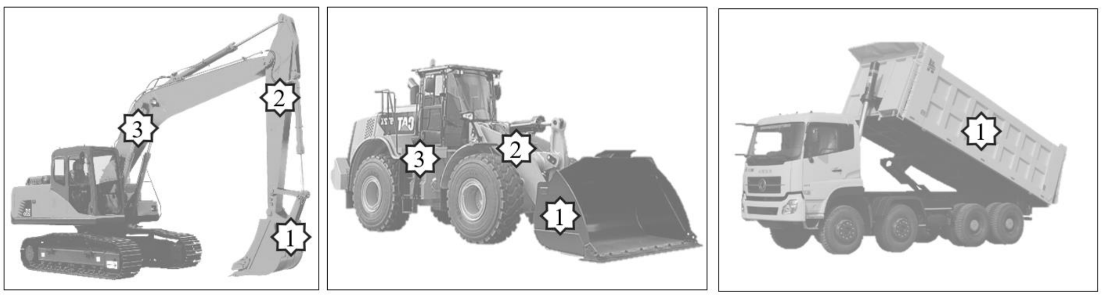


Figure 2.3. Location of IMUs on equipment

For the excavator, IMUs are attached to the bucket, stick, and boom. For the front-end loader, IMUs are attached to the bucket, boom, and cabin. For the dump truck, the one IMU is mounted on the dump body. All IMUs logged the accelerometer and gyroscope data in the microSD cards. The operations are videotaped for the duration of data collection using a camera from a static position. After collecting the data from IMUs, the raw data is processed for further analysis.

2.4.2. Data processing

After collecting the raw data from sensors, several data processing techniques are applied to prepare the raw data to train the LSTM network. Major steps in data processing are noise filtering, interpolating missing data, and data segmentation.

2.4.2.1.Noise Filtering

A noise reduction filter named *Median filter* is applied to reduce the mechanical noise of the IMUs. Median filtering is often applied to smooth this type of noise (Gather and Fried 2002). Figure 2.4 shows a sample raw accelerometer with dashed blue lines and smoothed filtered data (obtained after filtering) with solid orange lines.

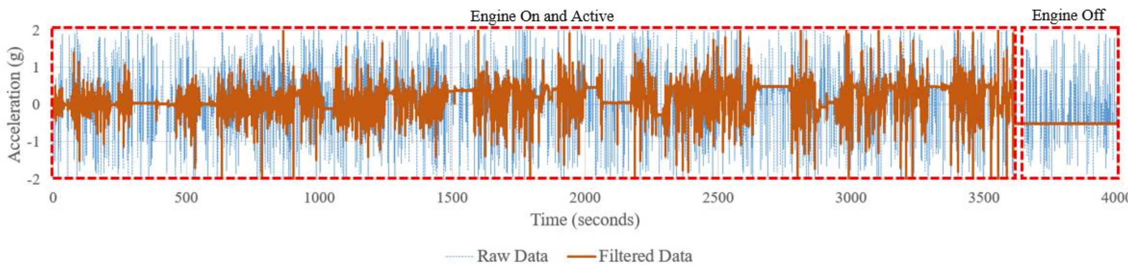


Figure 2.4. Raw and filtered IMU data

For purposes of illustrating the need for data processing, the last section in this figure represents the data corresponding to the idle state (engine off) of the excavator. In this section, the accelerometer data is expected to be a flat line, as there is no vibration on the equipment while the engine is off. Nevertheless, we see spikes in the data, which is essentially due to mechanical noise in the IMU. The solid line in Figure 2.4 shows filtered data as expected.

2.4.2.2.Interpolating missing data

The next phase of data processing eliminates any inconsistencies in the collected data. The IMU used in this study is capable of recording 80 data points per second which implies a sampling frequency of 80 Hz. However, the IMU can fail to record motion data at such a uniform time interval due to occasional freezing for a short time. To compensate for this missing data, the sensor records data at a higher than 80 Hz sampling frequency (Akhavian et al. 2015). In this research, the collected data with missing data points are processed into continuous uniform time series by removing redundant data and linearly interpolating the missing values. Figure 2.5 illustrates how the missing data are added using the linear interpolation method, after which it is segmented using the data segmentation technique.

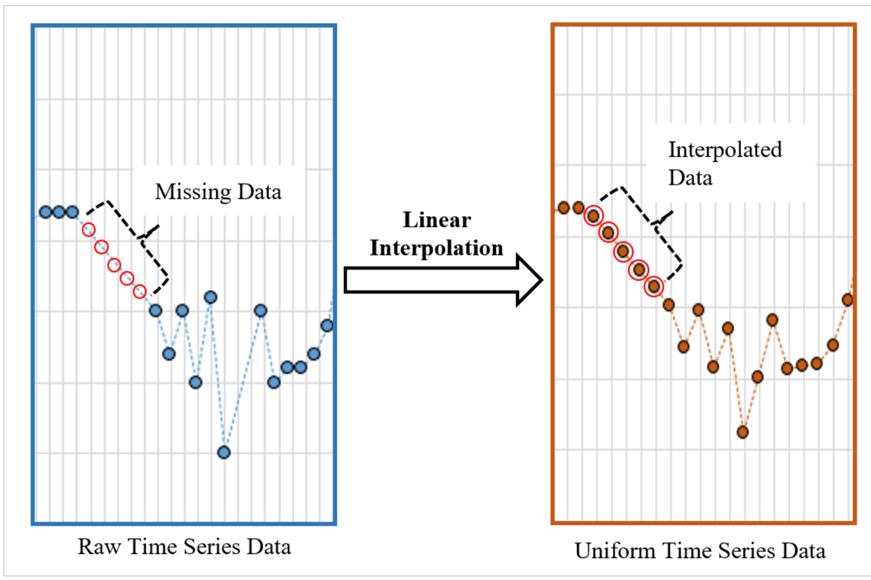


Figure 2.5. Linear interpolation of raw sensor data

2.4.2.3. Data segmentation

In this study, sliding window segmentation techniques are implemented. A fixed-sized window with 50% overlapping is considered since overlapping is useful when there is a transition between activities (Su et al. 2014). Figure 2.6 shows the sliding windows W_{m-1}, W_m, and W_{m+1} with 50% overlap between each of them on a one-dimensional data set. Several window sizes (1 second to 5 seconds) are selected to investigate the effect of window sizes on the classifier's performance.

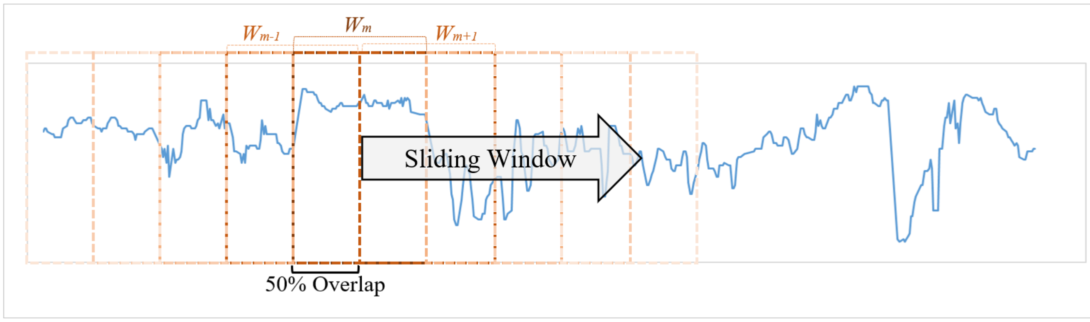


Figure 2.6. Data segmentation using a sliding window

2.4.3. Time-series data augmentation

This section describes data augmentation techniques that can be used to synthetically generate time-series data for deep learning. For image recognition applications, mirroring, scaling, cropping, rotating, etc. are legitimate augmentation techniques as minor changes due to these techniques do not alter the label of the image as they may happen in real-world observation. However, these label-preserving transformations are not intuitively recognizable for time-series IMU data. Factors that can introduce variability without altering the labels of the time-series data are random noise, sensor placements, and temporal characteristics of activities. To account for those factors *Jittering*, *Scaling*, *Rotation*, and *Time-warping* are implemented as shown in Figure 2 (Um et al. 2017). Figure 2(a) shows the raw data of one channel of the sensor.

2.4.3.1.Jittering

Jittering is implemented to simulate additive sensor noise. Each sensor has a different type of mechanical noise. Simulating random sensor noise increases the robustness of the training data against various types of sensors and their multiplicative and additive noises. White Gaussian noise is used in this study to add the jittering to raw training data. The effect of jittering on the test dataset is illustrated in Figure 2.7.

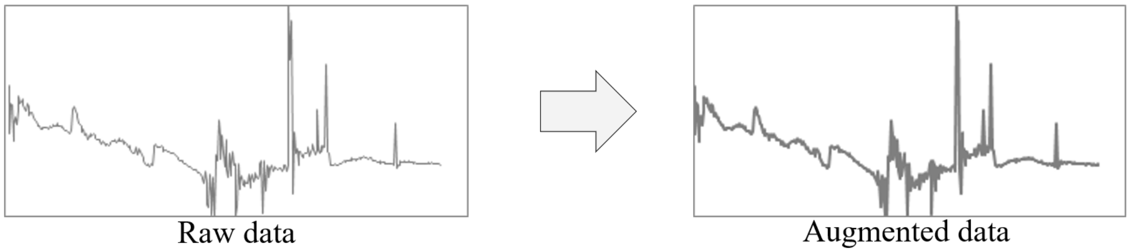


Figure 2.7. Data augmentation using jittering

2.4.3.2. Scaling

Scaling is another technique adopted in data augmentation which changes the magnitude of the raw data but preserves the labels. This variation is observed in situations wherein the dimensions of the implement to which the sensor is attached change, such as a change in length of excavator boom, etc. Scaling is implemented by multiplying the raw training data by a random scalar. Figure 2.8 shows an augmented dataset after applying scaling to the test data.

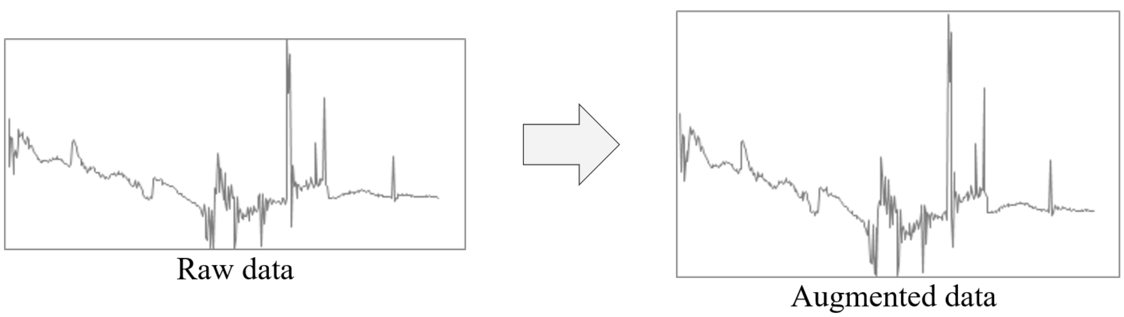


Figure 2.8. Data augmentation using scaling

2.4.3.3. Rotation

Rotation can be accounted for by introducing label-invariant variability of IMU sensor data when sensors are placed in the equipment with different orientations. For example, an upside-down placement of the sensor can invert the sign of the IMU readings without changing the labels as shown in Figure 2.9. Moreover, applying arbitrary rotation to the raw data can account for any minor changes in the sensor orientation because of vibration during data collection.

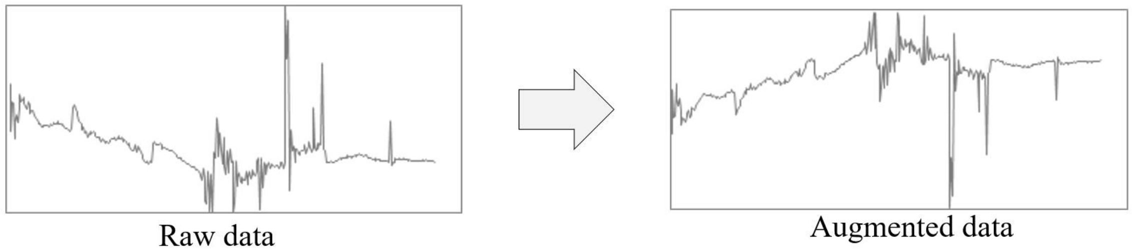


Figure 2.9. Data augmentation using rotation

2.4.3.4.Time-warping

Time-warping is a technique to generate synthetic training data for different temporal characteristics of equipment for a specific task. For example, a loading activity can be performed by an excavator with various operating speeds. Each activity was warped (i.e., stretched or shortened) with different warping ratios to account for this variability. A sample of warped data can be seen in Figure 2(e).

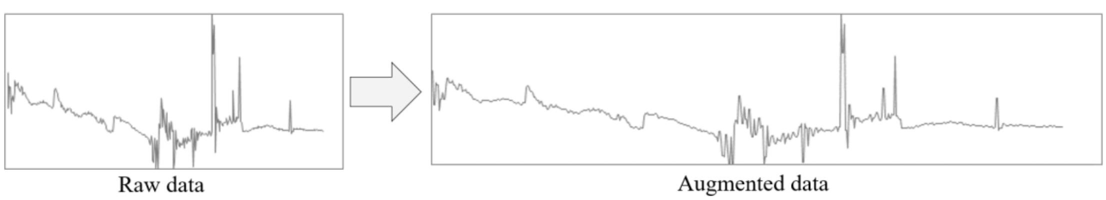


Figure 2.10. Data augmentation using time-warping

Each augmentation technique generated a 4-fold increase in augmented training data, resulting in a 16-fold increase in the number of the training data points. Four different signal-to-noise ratios were used for *Jittering* to generate 4-fold simulated data with different noise levels. For *Scaling*, scaler multiplication values of 0.8, 0.9, 1.1, and 1.2 were selected to generate 4-fold augmented data with slightly different magnitudes of the reading. Similarly, four different rotation factors were selected to change the sign of the IMU reading, which resulted in a further 4-fold increase of the training data. Finally, four warping ratios were selected to generate 4-fold synthetic data with various temporal lengths, preserving their labels.

2.4.4. Model training

An LSTM network is a type of recurrent neural network (RNN) that learns the long-term temporal dependencies between time steps of sequence data. The main components of an LSTM network for time-series classification are a *sequence input layer*, an *LSTM layer*, a *fully connected layer*, a *softmax layer*, and a *classification output layer* as shown in Figure 2.11 (Graves et al. 2013).

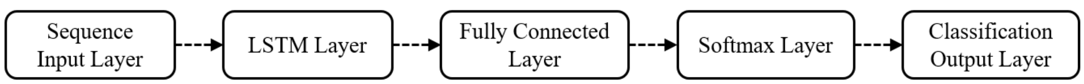


Figure 2.11. Main components of an LSTM network

Time-series data are used in the *sequence input layer* which inputs the sequences into the network. The *LSTM layer* learns long-term temporal dependencies between time steps of sequence data in terms of weight matrix and bias vector. The *fully connected layer* then multiplies the inputs by the weight matrix and adds the bias vector. Next, the *softmax layer* applies the neural transfer function to the input. Finally, the classification output layer computes the cross-entropy loss for multi-class classification problems with mutually exclusive classes. The *LSTM layer* is composed of LSTM units and a common architecture of LSTM units consists of a cell (i.e. *c*), input gate (i.e. *i*), output gate (i.e. *o*), and forget gate (i.e. *f*) as shown in Figure 4 (Graves et al. 2013).

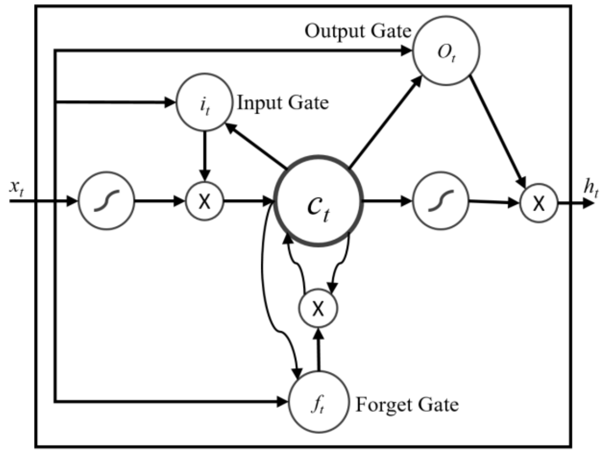


Figure 2.12. The architecture of a long-short term memory (LSTM) unit (Graves et al. 2013)

Each of these gates computes an activation, using an activation function of weighted sum. In the figure, i_t, o_t, and f_t represents the activations of input, output, and forget gate respectively at time step *t*, where x_t and h_t are the input and output vector of the LSTM unit. The three exit arrows from the memory cell *c* to 3 gates *i*, *o*, *f* denote the contributions of the activation of the memory cell *c* at time step *t*-1 (i.e. the contribution of c_{t-1}). In other words, the gates *i*, *o*, and *f* calculate their activations at time step *t* considering the activation of the memory cell *c* at time step *t*-1. The circle containing an X symbol in Figure 4 represents element-wise multiplication between its inputs. Also, the circle containing an S-like curve represents the application of an activation function (e.g. sigmoid function) to a weighted sum (Greff et al. 2015). In this study, the input vector *x* contains the time-series sensor readings of the IMUs. Unlike the traditional machine learning approach, where raw data are processes, segmented, and statistical features are extracted, the LSTM network can automatically learn high-level representative features containing the long-range temporal relationship between time steps. Thus, the augmented training data are used as input vectors in the LSTM to train the deep network. After the training phase, test data are used to evaluate the trained model.

2.4.5. Model evaluation

In this study, four common performance measures; accuracy, precision, recall, and F_1 score are used to measure the performance of the LSTM network, as well as to compare the LSTM network with ANN. The accuracy of the classification model can be calculated by dividing the number of correctly classified classes by the total number of classes as shown in Equation (2.1).

$$Accuracy = \frac{Number\ of\ correctly\ classified\ classes}{Total\ number\ of\ classes} \times\ 100\% \tag{2.1}$$

Precision and recall of the model are calculated to account for the cost associated with misclassification. Precision is the fraction of predicted positive instances (i.e., true positive + false positive) that are truly positive (i.e. true positive), while recall refers to the fraction of true instances (i.e. true positive + false negative) that are correctly predicted as positive (i.e. true positive) as shown in Equation (2.2) and (2.3).

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \times\ 100\% \tag{2.2}$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \times\ 100\% \tag{2.3}$$

While it is desirable to achieve high precision and recall value, it is often challenging to maximize both measures for a single classification model. Thus, F_1 score is calculated which is the harmonic mean of precision and recall, as shown in Equation (2.4).

$$F_1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{2.4}$$

Confusion matrix is also used to analyze the inter-class predictive performance of the trained model. Each row of the matrix represents the instances in an actual class where each column represents the instances in a predicted class.

2.5. Case study

To evaluate the proposed activity identification framework, three sets of time-series data were collected from an excavator, a front-end loader, and a dump truck. The collected sensor data were then separated into training and test datasets, where training dataset were used to generate more synthetic training data (i.e., data augmentation), and test data were used to validate the trained model. Several performance matrices were used to evaluate the performance of the LSTM network. Moreover, a comparative analysis was conducted to see the improvement resulted from the developed deep network compared to a traditional shallow network (i.e., ANN).

2.5.1. Data collection

Three case studies were performed by collecting three sets of IMU data from the construction site: from an excavator (Komatsu PC 300 LC), and a front-end loader

(Caterpillar 980G), and a dump truck (Bell Orion B60 E). Three IMU sensors were attached to the excavator (bucket, stick, and boom) and front-end loader (bucket, boom, cabin), while one IMU was attached to the dump body of the dump truck as shown in Figure 2.13.

Figure 2.13. IMU placements on the equipment during data collection

Each of the IMUs used in this study can record 3-axis accelerometer, and 3-axis gyroscope data (i.e., total 6 sensor readings per IMU). The three IMU sensors used for the excavator and the loader collected 18 channels of time-series data (i.e. three IMU multiplied by six channels) for each piece of equipment. Data were collected for approximately two hours for each piece of equipment. Given an average data capture frequency of 80 Hz, approximately 576000 data samples were collected for each channel of the sensor (e.g., accelerometer X) for each equipment. Furthermore, the operations of the equipment were videotaped for the entire duration of data collection to aid with data labeling and validation.

2.5.2. Data Labeling

Temporal labeling of data into different activities is vital in training the learning algorithm (Spriggs et al. 2009). The level of details (LoD) or the resolution for data labeling depends on the specific application. For example, for an environmental impact study of equipment, where the researcher may want to perform emission analysis during the idle phase of equipment, breaking down the activities into three classes (i.e., engine off, idle, and busy) can be proven sufficient. Considering this, this study investigates different LoDs and their impact on the classifier's performance. Although a higher level of LoD is more desired, the ML classifiers may not perform well given the relatively large number of classes. Moreover, as mentioned by Akhavian and Behzadan (2015), a higher level of LoD poses some inherent challenges. One such challenge is that the more are the levels, the less will be the number of training data points in some classes.

In this study, various LoDs were considered and investigated to see how this affects the performance of the trained algorithm. As an example, Figure 2.14 depicts the hierarchy of activities that are performed by the excavator.

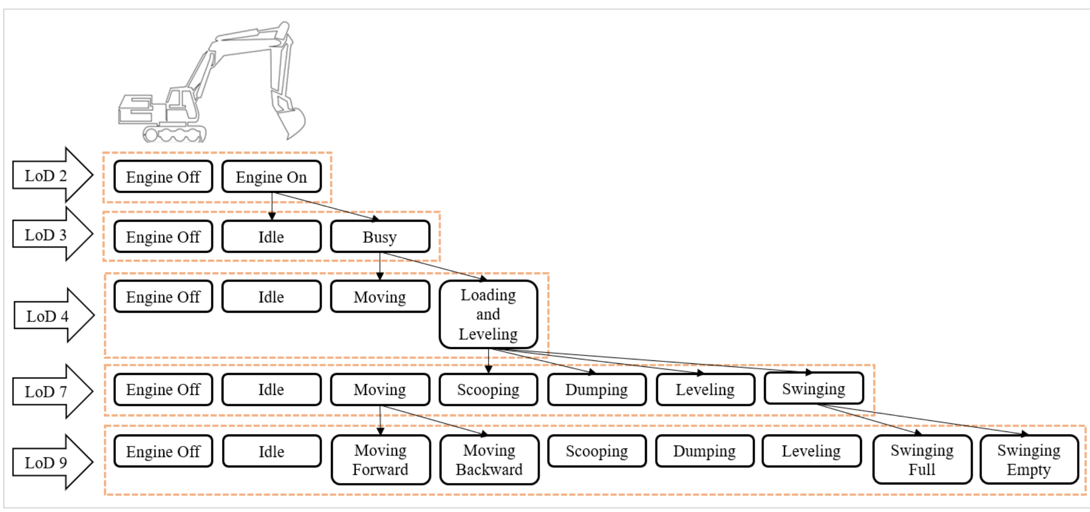


Figure 2.14. LoD in activity breakdown of the excavator

In the coarsest breakdown (LoD2), 2 classes are defined: *Engine Off* and *Engine On*. In the next level, *Engine On* activity is further divided into 2 more activities: *Idle* and

Busy. This process is continued, and the finest breakdown (LoD9) contains 9 classes: *Engine Off*, *Idle*, *Moving Forward*, *Moving Backward*, *Scooping*, *Dumping*, *Leveling*, *Swinging Full*, and *Swinging Empty*. Table 2.1 shows the lists of distinguishable activities for the excavator, front-end loader, and dump truck selected in this study.

Table 2.1. Selected activities for the equipment

Equipment type	**Name of the activities**
Excavator	*Engine Off, Idle, Scoop, Dump, Swing Loaded, Swing Empty, Move Forward, Move Backward, and Level Ground*
Front-end Loader	*Engine Off, Idle, Scoop, Raise, Dump, Lower, Move Forward Loaded, Move Backward Loaded, Move Forward Empty, Move Backward Empty*
Dump truck	*Dump, Idle, Travel*

If an excavator is loading, then the corresponding dump truck is also assumed to be loading. Moreover, the logical inference was used to extract haul and return activity for the dump truck. For example, if the truck is loading then after that it should be hauling, and if the truck is dumping, after that it should be returning. The reason for choosing a higher number of classes for excavator and front-end loader (compared to previous similar studies such as, (Ahn et al. 2015; Akhavian and Behzadan 2015; Mathur et al. 2015)) in this study was to test the robustness of the deep network even when the signal patterns of the IMU of different classes become more similar due to higher resolution. The next subsection discusses the implementation of the LSTM network.

2.6. Result and Discussion

From a model evaluation perspective, this study focuses on the following three questions:

1. Does deep learning outperform shallow learning?
2. Do data augmentation techniques improve the performance of the deep learning model?
3. Does data augmentation help to reduce the inter-class confusion of the LSTM network?
4. What is the best location of the equipment to attach the motion sensor?

This section is organized by first, summarizing all the performance measures in tabular forms. Then comparative performance analysis of ANN and LSTM network is conducted. The impacts of data augmentation techniques on the performance measures of both ANN and LSTM are investigated. Finally, a closer look at the confusion matrices of the LSTM network with and without data augmentation explores the inter-class confusion of the model.

2.6.1. Overall performance of the prediction model

As discussed in section 2.4.3, each of the augmentation techniques (e.g., jittering, scaling, rotation, and time-warping) were implemented using four different technical parameters. The volume of augmented training data is highest when each of the techniques was implemented four times, and lowest when each of the techniques was implemented just one time. In section 2.4.3, while each of the augmentation techniques was applied four times, it was mentioned to have 16- fold augmentation. However, for an easy understanding, the following sections in this chapter mention 4-

fold augmentation when each of the techniques was implemented 4 times (i.e. total 16-fold data increase). Similarly, if each of the techniques was applied 2 times, it is mentioned to have 2-fold augmentation (i.t. total 8-fold data increase). 5 different volumes of the training dataset (i.e., no augmentation, 1-fold, 2-fold, 3-fold, and 4-fold augmentation) were generated and the LSTM network was trained with each of them to evaluate the impact of data augmentation on the performance of the model. Figure 2.15 shows the overall performance of the trained LSTM model for three equipment using 4-fold data augmentation.

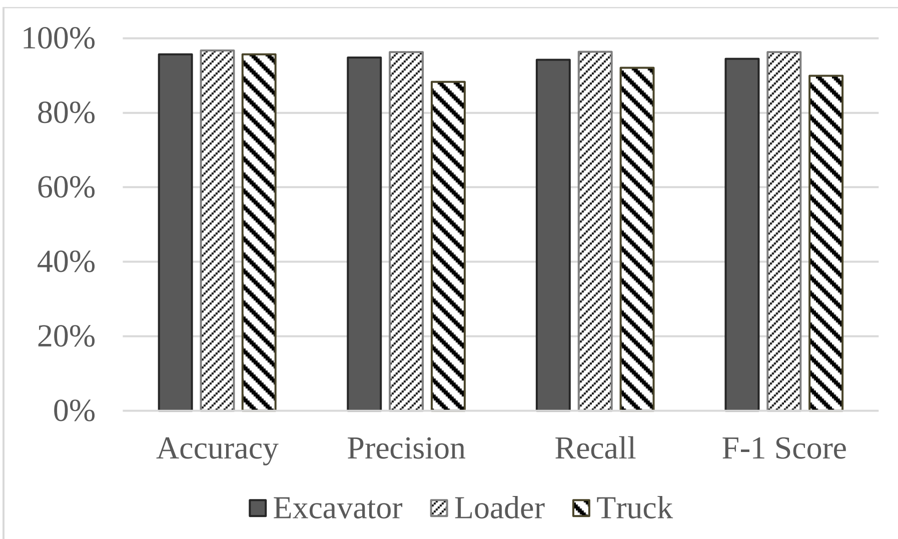


Figure 2.15. Overall performance of the LSTM model for all three equipment

The following activities were considered for the equipment for this analysis:

Excavator: *Handling, Loading, Idle, Traveling*

Front-end loader: *Engine Off, Loading, Idle, Moving*

Dump truck: *Idle, Traveling, Dumping*

The result shows that all the performance measures (i.e., accuracy, precision, recall, and F-1 score) are about 90%. However, precision for the dump truck is 88.3%. The idle and loading activity of the dump truck has a lot of overlap in terms of sensor data, and this could be a potential reason for comparative low precision value.

2.6.2. Performance of LSTM using data augmentation

At this stage of the analysis, the excavator and front-end loader was selected for further investigation due to their higher LoDs (i.e., more distinguishable activities as shown in Table 2.1). Testing the prediction algorithm with a higher number of activities will test the robustness of the model. In addition to training an LSTM network, an artificial neural network (ANN) was also trained for each of the training datasets to compare it with the LSTM network. This should be noted that the test data were separated before the data augmentation and used to evaluate the LSTM network and the ANN. This helps to compare the models and to evaluate the impact of data augmentation. Table 2.2 and Table 2.3 summarizes all the performance measures for both ANN and LSTM networks trained with the data from the excavator and the loader.

Table 2.2. Performance measures of ANN and LSTM for the excavator

	Accuracy		**Precision**		**Recall**		**F-1 Score**	
	ANN	**LSTM**	**ANN**	**LSTM**	**ANN**	**LSTM**	**ANN**	**LSTM**
No Aug.	62.2%	63.3%	50.5%	55.1%	54.0%	54.1%	51.3%	54.1%
1-fold	74.9%	94.0%	63.9%	91.3%	71.8%	92.7%	65.8%	91.9%
2-fold	78.1%	97.1%	69.9%	92.9%	73.3%	93.9%	70.9%	93.3%
3-fold	78.7%	97.9%	70.3%	95.9%	73.5%	97.8%	71.3%	96.7%
4-fold	79.9%	97.9%	70.7%	96.2%	74.8%	99.0%	71.3%	97.6%

Table 2.3. Performance measures of ANN and LSTM for the loader

	Accuracy		**Precision**		**Recall**		**F-1 Score**	
	ANN	**LSTM**	**ANN**	**LSTM**	**ANN**	**LSTM**	**ANN**	**LSTM**

No Aug.	48.8%	59.7%	36.6%	52.6%	47.1%	54.7%	35.4%	52.4%
1-fold	62. 6%	78.7%	51.1%	75.8%	61.9%	77.9%	51.8%	76.6%
2-fold	63.8%	94.1%	52.7%	93.3%	61.6%	93.1%	53.7%	93.2%
3-fold	64.1%	95.4%	54.2%	93.7%	63.1%	95.1%	55.5%	94.4%
4-fold	66.1%	96.7%	56.1%	96.3%	64.7%	96.4%	57.1%	96.3%

The left-most column of both the tables shows the amount of data augmentation. *No Aug.* represents only the raw training data (i.e., no augmentation), and *4-fold* represents augmenting raw training data 4 times with each of the augmentation techniques. Each of the performance measures is listed side-by-side in the tables for an easy comparison of ANN and LSTM. These two tables demonstrate two major findings:

1. The deep learning network (i.e., LSTM) improves the performance of the prediction model compared to shallow networks (i.e., ANN).
2. Data augmentation improves the performance of the LSTM network.

These two tables are deconstructed with different types of data visualization (e.g., bar chart, line graph, etc.) for more detailed analysis.

2.6.3. Amount of augmentation and prediction performance

This step of the analysis explores how different amounts of augmentation impacts the performance of the prediction model.

Figure 2.16 and Figure 2.17 plot the F-1 score for both ANN and LSTM network for the excavator, and the loader respectively. The x-axis shows the amount of augmented training data.

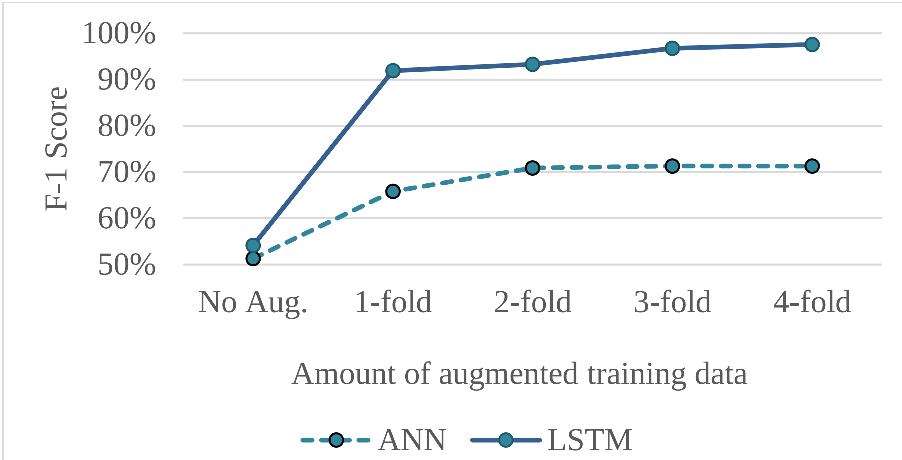


Figure 2.16. F-1 score with different amounts of augmented training data for the excavator

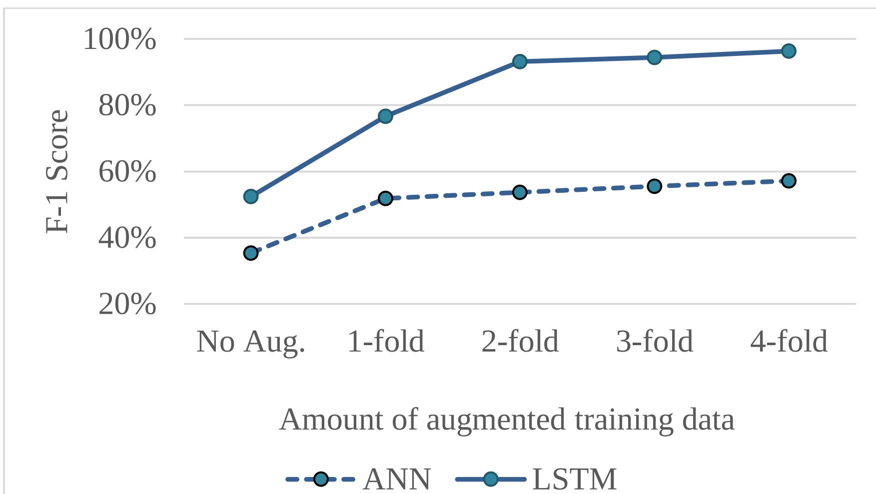


Figure 2.17. F-1 score with different amounts of augmented training data for the loader

The LSTM performance increases with an increase in training data at a higher rate than ANN. For example, the F-1 score of LSTM and ANN is 54.1% and 51.3% (i.e., a difference of 2.8%) respectively for the excavator when trained with only raw training data (i.e., no augmentation). However, when the amount of training data is increased to 1-fold, the difference in F-1 score is 26.1% (i.e., 91.9% for LSTM and 65.8% for ANN). A similar characteristic is seen in Figure 2.17 as well. This exponential increase in performance supports the argument that deep networks outperform shallow networks with larger training datasets (Schindler et al. 2016). The performance measures for both ANN and LSTM are visualized in Figure 2.18 (excavator) and Figure 2.19 (loader).

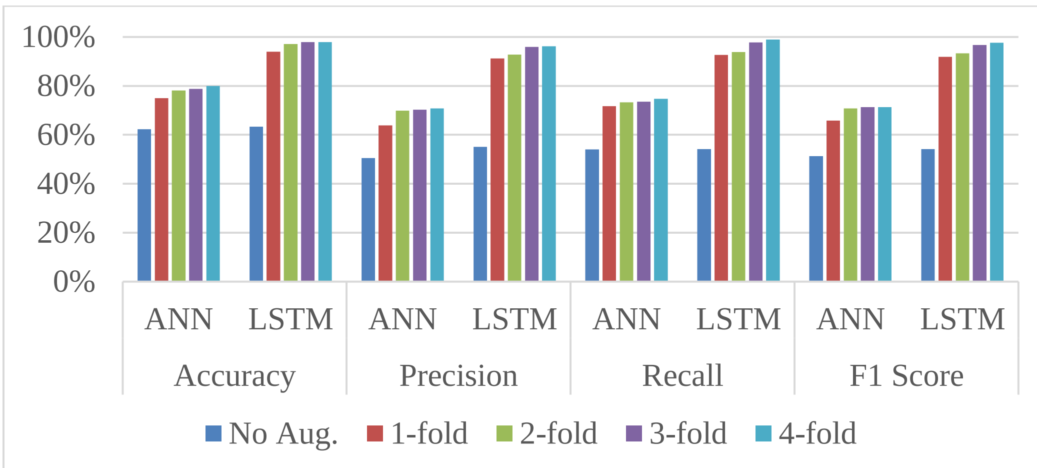


Figure 2.18. Performance measures of ANN and LSTM for the excavator

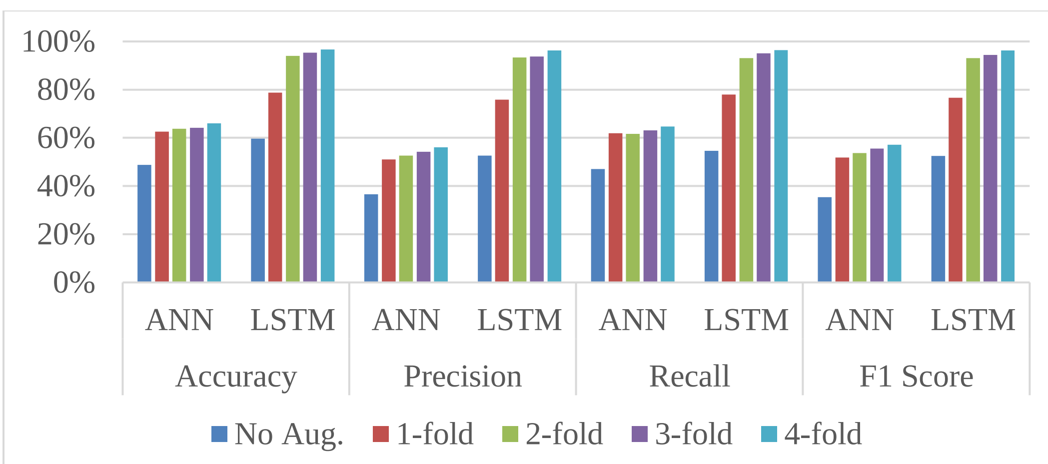


Figure 2.19. Performance measures of ANN and LSTM for the loader

From both the figures, a positive impact of data augmentation on the each of performance measures can be observed. Accuracy, precision, recall, and F-1 score increase with each phase (e.g., 1-fold, 2-fold, etc.) of data augmentation. Specifically, significant improvement is noticed from *No Aug.* to *1-fold* augmentation. For example, precisions of the LSTM network for the excavator are 55.1%, 91.3%, 92.9%, 95.9%, and 96.2% for *No Aug*. *1-fold*, *2-fold*, *3-fold*, and *4-fold* augmentation. This supports the argument that data augmentation techniques can generate synthetic training data by infusing variations to the raw data without altering their classes. We see an overall improvement of 34.6% in accuracy, 41.1% in precision, 44.9% in recall, and 43.5% in F-1 score for the LSTM network after applying the data augmentation for the excavator. Similarly, the LSTM network for the loader illustrates the significant improvement of the performance measures after introducing data augmentation.

2.6.4. Inter-class error using the confusion matrix

Even though accuracy, precision, recall, and F-1 score represent the overall performance of the LSTM network, they do not provide any information on how instances are misclassified. Thus, confusion matrices are introduced to identify the classes that are misclassified and confused with other classes. Figure 2.20 and Figure 2.21 show the confusion matrices with and without data augmentation for the excavator and the loader respectively. Each row represents the actual classes, and columns represent predicted classes. The green diagonal cells in these figures represent correctly classified classes, where all other cells show the misclassified classes.

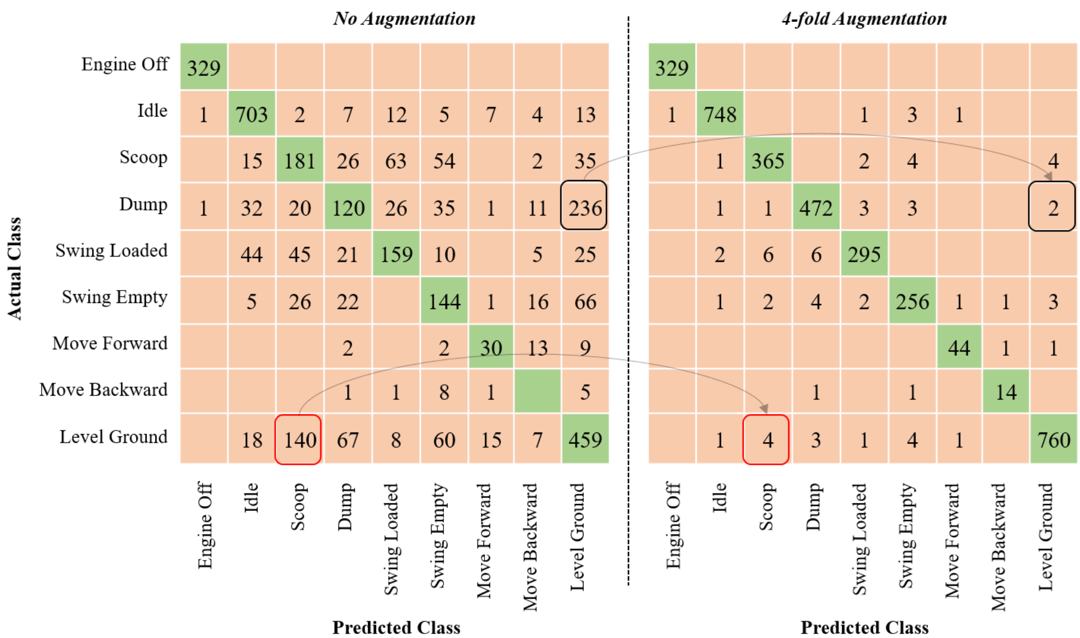


Figure 2.20. Confusion matrix of LSTM network for the excavator

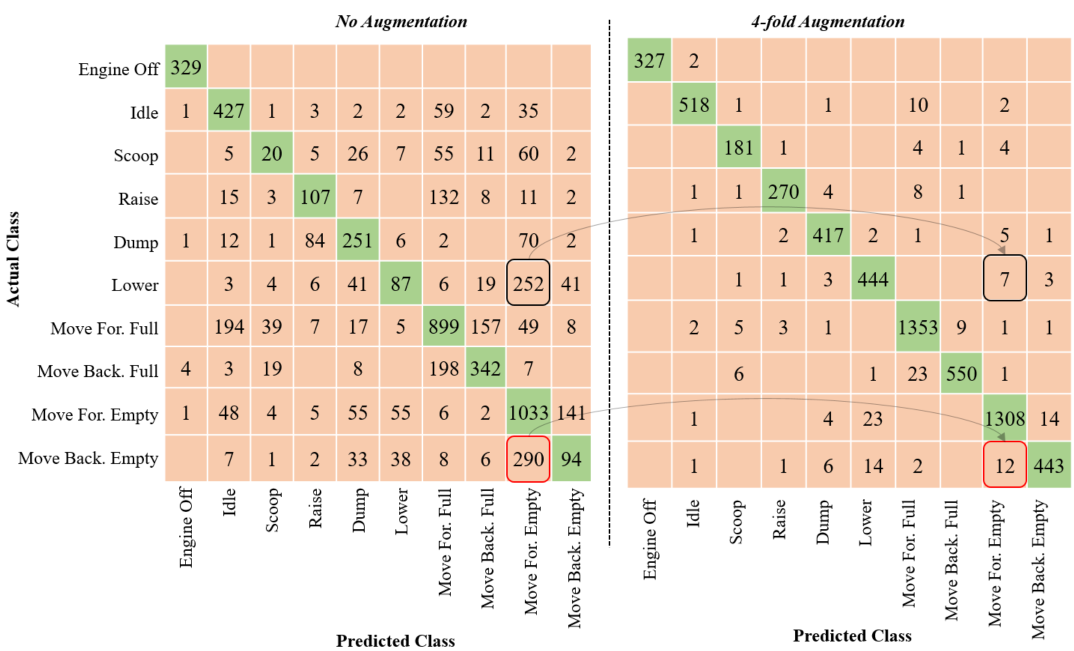


Figure 2.21. Confusion matrix of LSTM network for the loader

From Figure 2.20, we see that the top two misclassified activities for the excavator are *Dump* and *Level Ground* before data augmentation. *Dump* is confused 236 times with the *Level Ground* (highlighted by black border), where *Level Ground* is confused 140 times with the *Scoop* (highlighted by a red border). During the *Level Ground* activity, the excavator was picking up a small amount of soil (similar to *Scoop*) and dumping them in close proximity (similar to *Dump*). Thus, the confusion among *Level Ground, Dump, and Scoop* can be explained by the similar signal patterns generated from the IMUs. However, after applying the augmentation these confusions are noticed to be reduced 236 to 2 times for the *Dump* activity and 140 to 4 times for the *Level Ground* activity. We see a similar situation in Figure 2.21, where 252 instances of confusion between *Lower* and *Move For. Empty* has reduced 7 instances, and 290 instances of confusion between *Move Bac. Empty* and *Move For. Empty* is reduced to 12 instances. A significant reduction of misclassified instances is noticeable in the figures after applying data augmentation. This supports the argument that introducing slight variations (i.e., data augmentation) to the raw training data to generate more data helps better generalization of the deep network.

2.6.5. Window sizes and prediction performance

Several sensitivity analyses were performed to check the effect of different window sizes on the performance of the prediction model. For this analysis, the excavator was chosen as the equipment, and ANN was chosen as the classification model. The reason behind choosing ANN, over LSTM for this analysis is that LSTM generally takes a long time to train. As this stage requires training models for different window sizes, computationally it was advantageous to train the ANN. Moreover, as this is a comparative analysis, ANN and LSTM are expected to perform with a similar pattern. Figure 2.22 shows the effect of window sizes on the prediction performance. Four different levels of details were considered and 20 different window sizes (from 1 second to 20 seconds) were used to train the model. Each time, a 50% overlap was used with the sliding window technique as described in section 2.4.2.3.

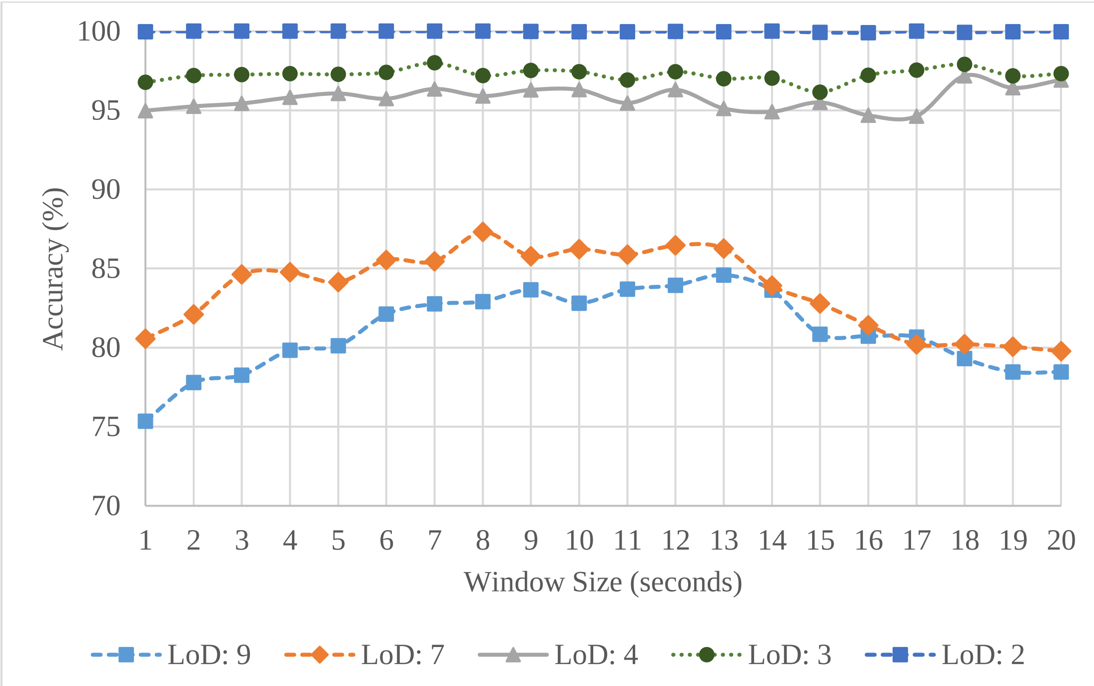


Figure 2.22. Effect of window size on prediction performance for the excavator

It can be seen from the figure that the accuracies of the model are similar for lower LoDs (LoD: 2, LoD: 3, and LoD: 4) with window sizes. However, for higher LoDs (i.e., LoD: 7 and LoD: 7), the accuracy shows an upward trend initially when the window sizes are increasing, and after a peak, the accuracy tends to decrease with the increase of window size.

2.6.6. Sensor placement and prediction performance

For this analysis, the excavator was selected as the equipment, and ANN, K-nearest neighbor (KNN), and support vector machine (SVM) were chosen as the classification models. Different combinations of the sensor placement and training models were used in this step to analyze the placement of the sensors to optimize the performance of the prediction model. While previous efforts have found promising results in equipment activity recognition using a single smartphone-embedded sensor mounted inside the equipment's cabin (Ahn et al. 2015; Akhavian and Behzadan 2015; Mathur et al. 2015), this research utilizes three IMU sensors attached to three different implements of an excavator to leverage the motion of each moving part of the equipment. As shown in Figure 2.3 three IMUs were attached to the bucket, stick, and boom of the excavator. To describe how these sensor placements affect the prediction performance they were named IMU #1 (in the bucket), IMU#2 (in the stick), and IMU#3 (the boom). The impact of the sensor placement on the equipment is summarized in Figure 2.23.

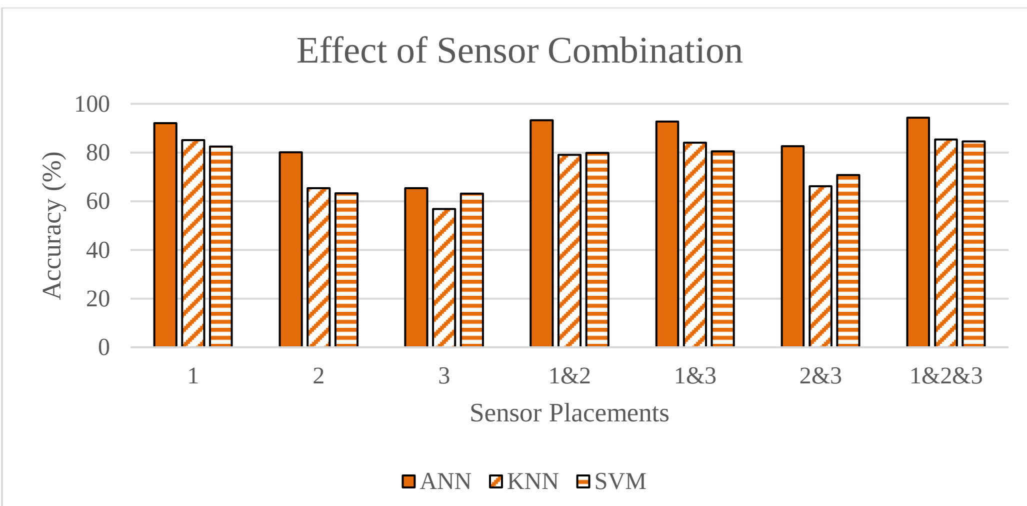


Figure 2.23. Impact of placement and combination of IMUs on classifiers

Figure 2.23 illustrates that fusing data from all three IMUs results in the highest performance for all three classifiers. If data from a single IMU is used for training the model, accuracy is highest for IMU#1 (attached to the bucket), and lowest for IMU#3 (attached to boom). For example, the accuracy of KNN using IMU#1, #2, and #3 are

85.1%, 65.3%, and 56.8% respectively while all three IMUs produce an accuracy of 85.4%. Moreover, when only two IMUs are used, accuracy is comparatively higher when IMU#1 is in use. Accuracy using combination of IMU #1&#2, IMU #1&#3, and IMU #2&#3 are 79%, 84.1%, and 66.2%, respectively. The performance of the prediction model using a single IMU is shown in Figure 2.24.

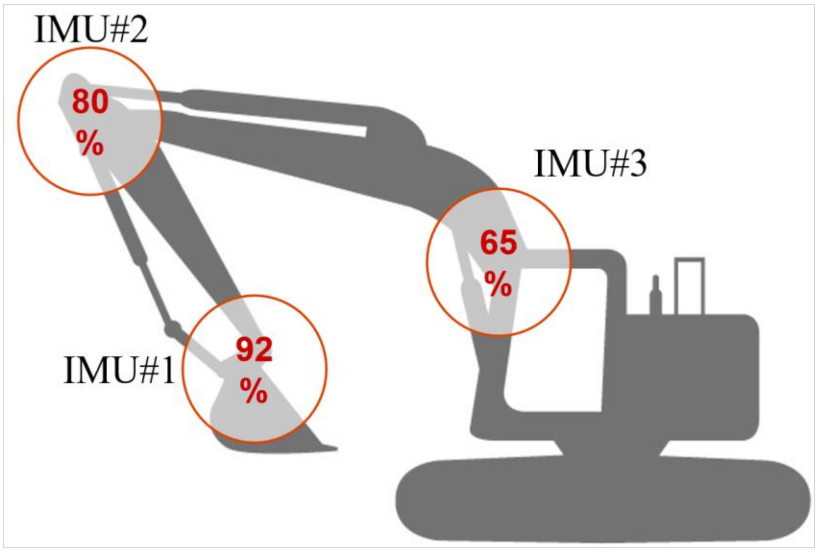


Figure 2.24. Accuracy of the model using a single IMU sensor

As IMU#1 is attached to the bucket of the excavator, it can be concluded that motion data of the bucket contributes more to understanding the operational activity of the excavator, than motion data of the stick or the boom. This section of the results indicates that using data from multiple IMUs attached to different implementations of the excavator improves the accuracy rather than using any sensor individually. Moreover, the bucket, which is the end effector of an excavator, is the best location for a single sensor to be attached to identify equipment's activity. As the bucket has the highest degree of freedom in terms of movement, it has the highest level of motion data. Vice versa, the boom has the lowest degree of freedom in terms of movement. Thus, it can be concluded that the location of the equipment with the highest level of movement is the best place to attach the sensor.

2.7. Major findings and research contributions

The proposed framework in this chapter consists of two major components: an LSTM deep learning-based activity recognition framework for construction equipment, and time-series data augmentation techniques to generate synthetic training data. The major findings of this chapter are:

- The deep learning network outperformed the shallow network in terms of accuracy, precision, recall, and F-1 score. As opposed to the traditional machine learning classification algorithms, the LSTM network contains long-term temporal dependency of the training data between consecutive time steps
- The LSTM network eliminated the necessity of manual feature extraction, which is limited to human domain knowledge. Instead, the deep network automatically learned high-level representative features from the raw training data.
- Implementation of data augmentation reduced the amount of training data from the construction site. This improves the practicality of such classification techniques for temporary and transient construction operations.
- The location with the highest degree of movement is the best place to attach the sensor for activity identification purposes.

The major contributions of this study to the body of knowledge and practice are:

- The deep learning-based automated activity identification framework provides means for future researchers and practitioners to track heavy equipment activities more accurately and reliably compared to traditional (shallow network-based) approaches.
- The proposed framework is implementable irrespective of weather conditions and site layouts compared to vision-based approaches.
- The combination of the learning model (i.e., LSTM network) and the data augmentation technique reduces the manual effort both in terms of field data collection and manual data processing (i.e., feature extraction).
- Determination of optimal sensor placement will help future researchers and original equipment manufacturers (OEMs) to deploy sensors more efficiently.

2.8. Conclusions and future work

An automated, real-time, and reliable activity recognition framework for construction equipment provides a foundational platform to monitor and assess productivity, safety, and environmental impact on construction sites. Towards this end, this chapter provides a deep learning-based activity recognition framework for construction equipment using multiple IMUs attached to different articulated elements of the equipment and utilizing the capabilities of the LSTM network. Moreover, several time-series data augmentation techniques were developed and implemented to generate synthetic training data, reducing the necessity of a large volume of data collection from the construction site.

This chapter showed that the deep learning approach (i.e., LSTM network) outperforms the shallow network (i.e., ANN) to identify equipment activities more accurately and reliably. The data augmentation techniques developed and implemented in this research shows the ability to correctly simulate real-world training dataset. This helps rigorous training of deep networks without collecting a large amount of training data from the field. Moreover, the implementation of data augmentation helps to reduce inter-class confusion of the LSTM network. Analysis of sensor placement shows the location with the highest level of movement is the best place to attach the sensor. Successful application of the proposed framework has the potential to transform the way construction operations are currently being monitored. By executing the proposed framework in real construction sites, construction operations can be continuously monitored and assessed in real time, relieving construction companies from the time-consuming and subjective manual method of analyzing construction operations.

The proposed framework can identify activities of three specific construction equipment (e.g., excavator, front-end loader, and dump truck). The designed methodology will be broadened to cover other types of equipment to ensure its robustness. In the future, the activity level information gathered from this study will be used for productivity analysis, safety analysis, and fuel use analysis techniques to support better decision-making and control methods. In addition to that, the proposed framework will be extended for human workers in construction sites using wearables to enable productivity and safety applications.

CHAPTER 3: DYNAMIC PRODUCTIVITY ESTIMATION FOR EARTHMOVING OPERATIONS USING SIMULATION AND BAYESIAN UPDATING

3.1. Introduction

Once the activities of the equipment can be automatically identified as explained in the previous chapter (i.e., Chapter 2), this chapter investigates how the outputs of the study can be used in the decision-making process. One of the suitable and available methods to aid decision-making based on productivity estimation for such operations is a discrete-event simulation (DES). However, the typical input data and logic for DES models are based on a-priori assumptions related to project conditions. Thus, as the project progresses, the models lose their validity as they may not reflect the updated project site conditions, resulting in unreliable outputs for decision-making. One method to overcome this limitation is to update simulation model inputs with the latest data collected from the project site – a process known as near real-time simulations (NRTS).

As a repetitive long-term activity, earthmoving operations provide a very suitable opportunity to perform NRTS by fine-tuning the activity duration inputs from the real world. This chapter provides a framework that integrates automatically collected field data from IMUs mounted on heavy equipment and processes them for use as updated input for simulation models to estimate dynamic productivity which presents the operation realistically. Activities (e.g., loading, hauling, etc.) are identified from the collected data and the cycle times are calculated using machine learning methods. The cycle time distributions are updated using Bayesian updating and used as inputs in the simulation model for more accurate productivity prediction. A real-world earthmoving site is used in the case study to validate the framework. This framework will help the decision-making process by realistically reflecting the field condition using the most up-to-date information in the simulation model.

3.2. Background

One of the major challenges in simulation modeling for productivity estimation is that they typically do not adapt to the changing conditions of the real world and the model violates the initial assumptions used for estimation. Typically, the inputs for simulation modeling are statistical data from past projects and the personal experiences and subjective judgment of decision-makers. However, as the project progresses, an additional layer of uncertainties in the form of changing site conditions, equipment breakdowns, weather delays, etc. are introduced into the project which makes the initial models unreliable for continued decision-making. This lack of complete information at the start of operations can lead to incorrect fleet sizing, resulting in lost productivity and time and cost overruns. One method of overcoming this inherent limitation of a-priori estimation is to constantly update the simulation model in real time using data that is available from real-world operations. Such a method can lead to data-driven decision-making for construction operations and is referred to as near real-time simulation (NRTS) (Akhavian and Behzadan 2012b; Louis et al. 2014; Vahdatikhaki et al. 2013; Vahdatikhaki and Hammad 2014).

This chapter builds upon the previous chapter (i.e., Chapter 2) as well as previous research in NRTS methods and presents an integrated framework for data-driven decision-making for construction operations with a focus on heavy civil earthmoving operations. The framework presented in this chapter starts by collecting motion data from earthmoving equipment from a real-world earthmoving site and convert those motion data into activities using deep learning techniques. Once activities are identified, their durations are fitted to distributions using Chi-square goodness-of-fit test. Initial predictions from a base DES model are updated by providing the most recent activity data as input to the model using Bayesian updating techniques. The updated distribution is then used as input parameters for the simulation model and an updated productivity estimation is obtained. By continuing this process, the simulation model accurately represents the actual activities on the site as the project progresses. A real-world earthmoving site was used as a case study to validate the framework. The results obtained show that the actual productivity on-site was different than the initial assumption and thus project schedule needed to be updated accordingly. This framework shows the feasibility of using dynamic simulation modeling to predict project outcomes more accurately and reliably for earthmoving operations.

The two major components of this study are activity identification and simulation modeling for heavy civil operations. This section provides an overview of the state of art in these two areas to identify the research gaps and define the point of departure, research goal, and objectives for this research.

3.2.1. Activity identification for heavy civil operations[2]

3.2.2. Simulation modeling and analysis for heavy civil operations[3]

3.3. Research gaps and point of departure

Based on the literature review conducted, the following research gaps are identified and targeted in this chapter:

(1) A dynamic estimation framework is missing for the heavy civil operation which uses the most recent field data and near real-time simulation modeling.

(2) Even though there is literature on using machine learning techniques to identify equipment activities, there is no significant work that integrates activity identification with simulation modeling to aid the decision-making process.

Considering the gaps in knowledge, the goal of this research is to develop a framework for dynamic productivity estimation for heavy civil operations by using up-to-date field data to facilitate better decision-making. This research goal is accomplished in this chapter through the pursuit of these specific research objectives:

(1) Utilize the activity identification framework developed in Rashid and Louis (2019b)

(2) Develop a DES model for earthmoving operation and update the model by refining cycle times acquired from the activity identification using the Bayesian updating method.

---

[2] Refer to Section 1.1.1 for the literature review of this part.
[3] Refer to Section 1.1.2 fo the literature review of this part.

3.4. Methodology

The methodology developed in this chapter follows the system architecture shown in Figure 3.1.

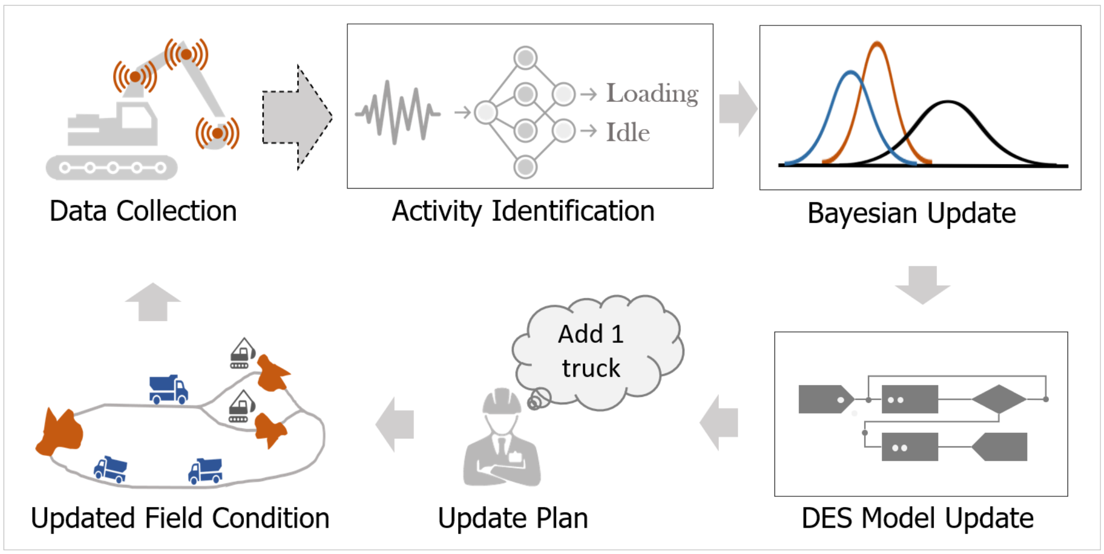


Figure 3.1. The system architecture of the proposed framework

Equipment activities are identified using IMU data collected from the field and deep learning techniques. The duration of each activity is calculated and fitted to the distribution using Chi-square goodness-of-fit. As the project progresses, the prior distributions are updated using the Bayesian updating technique and these updated distributions are used as a new input to the simulation model. A new prediction is generated from the model and the plan is updated based on the new prediction. The three major methodological steps; *activity identification*, *Bayesian update*, and *DES model update* are expanded in the following subsections.

3.4.1. Activity identification

Motion data (e.g., acceleration, orientation, etc.) from different articulated body parts of the equipment are captured using IMUs. These data are processed to eliminate noises and inconsistencies. Data augmentation techniques are used to generate synthetic data capturing the inter-class dependencies among different activities (e.g., loading, hauling, etc.). Then a deep learning technique, long short-term memory (LSTM)- based recurrent neural network (RNN) is trained and tested to identify activities of the equipment. Details of this activity identification framework are presented in (Rashid and Louis 2019b). Four major activities are identified in this study, which are essential to developing the DES model, and they are *Loading*, *Hauling*, *Dumping*, and *Returning*. Durations of each of those activities are calculated and fitted to distribution using Chi-square goodness-of-fit method. These duration distributions are used as inputs to the DES model.

3.4.2. Bayesian updating technique

The inputs (i.e., duration of the activities) to the simulation model are assumed to be continuous with an underlying probability density function (PDF). The prior assumptions are updated using Bayes' theorem when new duration data are calculated from the activity identification model. The Bayesian updating can be expressed with Equation (3.1)

$$f''(\theta) = kL(\theta)f'(\theta) \tag{3.1}$$

Here, $f'(\theta)$ is the prior distribution, which is revised to the posterior distribution $f''(\theta)$, θ is the random variable for the parameter of a distribution, k is the normalizing constant $k = \left[\int_{-\infty}^{\infty} L(\theta)f'd\theta\right]^{-1}$, and $L(\theta)$ is the likelihood of observing the experimental outcome assuming a given θ. The initial distribution assumption for the duration parameter is updated using the more recent observed data. Thus, the judgments and observational data are systematically combined as posterior distribution is obtained from both prior distribution and likelihood function. If the prior distribution is a conjugate of the distribution of the underlying random variable, the posterior distribution can be calculated in the same mathematical form as the prior (Ang and Tang 1975). Equation (3.2) and (3.3) show the updating process of normally distributed data.

$$\mu'' = \frac{\sigma^2 \mu' + n\sigma'^2 \mu}{n\sigma'^2 + \sigma^2} \tag{3.2}$$

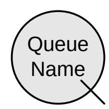
$$\sigma'' = \sqrt{\frac{\sigma^2 \sigma'^2}{n\sigma'^2 + \sigma^2}} \tag{3.3}$$

Here, μ'', μ', μ are posterior, prior, and sample mean respectively, and $\sigma'', \sigma', \sigma$ are posterior, prior, and sample standard deviation, n is the sample size. When more data are obtained after the first update, the updating process can be done successfully using these two equations. The posterior statistics from the previous stage become the prior distribution for the next updating stage. Based on these updating techniques, the duration parameter of the DES model is updated using a predefined frequency, thus, making the model more accurate and reliable.

3.4.3. DES modeling of heavy civil operations

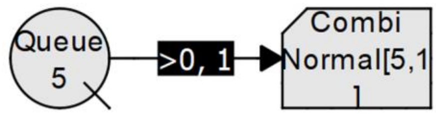
A simulation model of the earthmoving operation is developed. The model is built using a discrete event simulation software named jStrobe (Louis and Dunston 2016b). This is a construction-oriented discrete event simulation package that allows the user to model earthmoving operations as a chronological sequence of events with the help of a user-friendly graphical interface. jStrobe allows the user to model a variety of different operations using different levels of details. The major two components of the DES model are nodes and links. The nodes used in the DES can be of the following types:

Queues are nodes holding the resources until any activity draws them away. Attributes of the queues can be accessed by the modeler such as, the number of resources residing in the queue, the total amount of resources that entered the queue, and the average waiting time of each resource in the queue. A queue is represented by the following symbol in jStrobe:



Activities are nodes that represent the tasks to be performed in the model using the required resources. Each activity has a time duration that follows a distribution, such as normal, triangular, gamma, etc. Activities can be of the following two types: Combi and Normal. Combis are activities whose startup depends on certain conditions being met. On the other hand, Normals are activities that start starts as soon as the preceding activity is completed. The following two symbols represent the Combi and Normal activity.

The different nodes in the network are connected by links. The resources from one node to another is passed through the links by meeting certain conditions. The following figure shows a link that will pass one resource from the Queue to Combi when the number of resources in the Queue is greater than zero.



3.5. Case study and results

To evaluate the proposed methodology a real-world earthmoving site was selected. This was a highway extension project located in Oregon. The earthmoving operation was mainly carried out using five dump trucks and two excavators. There were two loading sites, each having one excavator, and one dumping site. Thus, the trucks need to decide upon which excavator to go to after dumping their load. Figure 3.2 shows a schematic layout of the site.

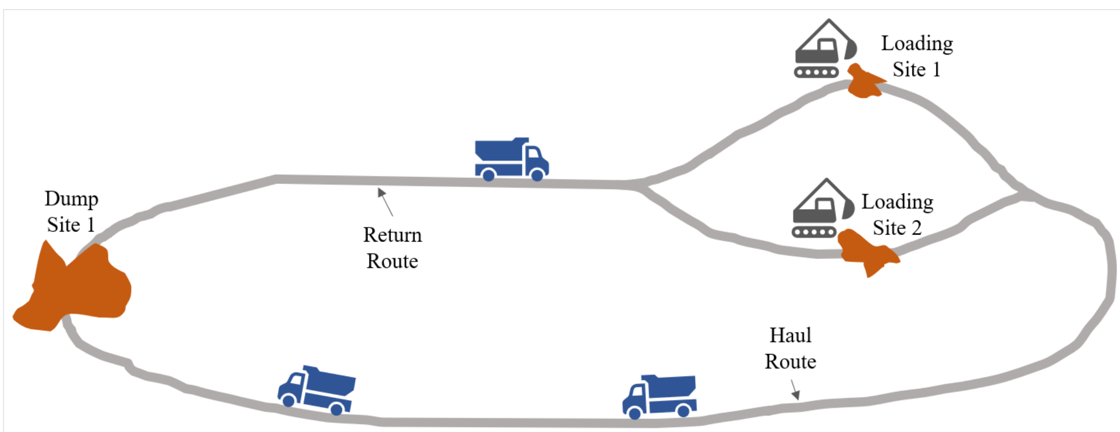


Figure 3.2. Schematic layout of the earthmoving site

3.5.1. Data collection

IMU sensors were attached to the equipment using a plastic box with a magnetic bottom. For excavators, IMU sensors were attached to the bucket as previous studies showed bucket is the best place to maximizes the activity identification accuracy (Rashid and Louis 2020a). GPS sensors were also attached to the trucks to record their movements. The activities were recorded using a video camera. Unfortunately, one sensor fell off a truck, and one sensor malfunctioned. Thus, data from three trucks and two excavators were available for further analysis. Figure 3.3 shows the sensor attachment to the dump truck and excavator.

Figure 3.3. IMU attachment to dump truck and excavator

Six-channel data (3-axis accelerometer and 3-axis gyroscope data) were collected from the IMU sensors to be used in the LSTM network for activity identification. The raw data from the accelerometer shows specific and distinguishable patterns for different types of activities. Figure 3.4 shows x-axis data of the accelerometer for one truck. We can see that the cyclical activities (i.e., load, haul, dump, return) of the dump trucks are highly noticeable from this one-channel data.

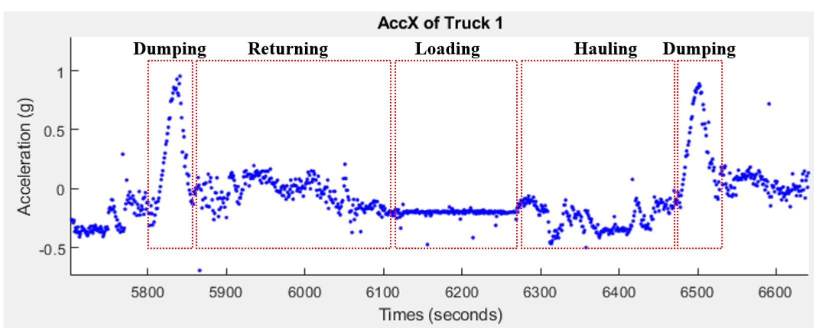


Figure 3.4. Distinguishable patterns of accelerometer data for dump truck

3.5.2. Cycle times estimation using activity identification

The collected data were used to train an LSTM network to identify the activities following the steps described in (Rashid and Louis 2019b). The duration of all the activities was calculated once the activities were identified accurately. Figure 3.5 shows the duration of different activities of the dump trucks. The average duration for *Loading, Hauling, Dumping, and Returning* are 2.81 minutes, 15.03 minutes, 0.83 minutes, and 15.53 minutes, respectively. These durations will be the primary input parameter of the DES model as a distribution (e.g., normal distribution, gamma distribution, etc.). The distributions were fitted using the Chi-Square Goodness-of-Fit test, which tests if a sample of data came from a specific theoretical distribution. This test groups the data into bins, then calculates the observed and expected counts for those bins and computes the chi-square test statistics. All four activities were fitted to normal distribution. Figure 3.6 shows the histogram and probability density of the loading times.

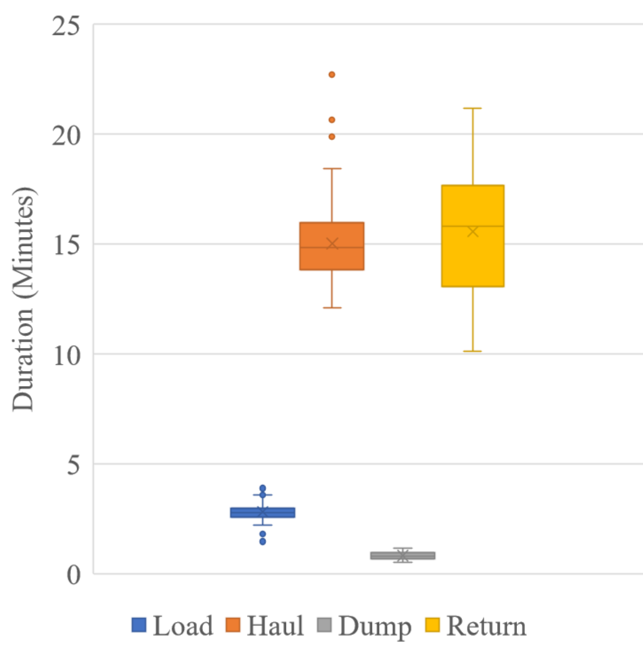


Figure 3.5. Cycle time durations of dump trucks

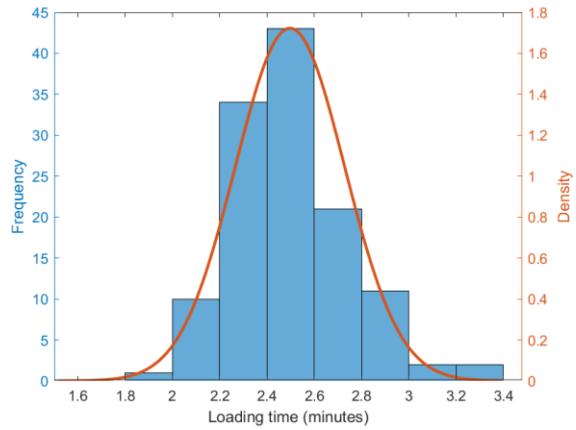


Figure 3.6. Distribution of loading time

The calculated means and standard deviations for all four activity durations are shown in Table 3.1. The return route for the trucks was longer than the hauling route which is reflected in the data. These values were assumed to be the initial input parameters for the base DES model.

Table 3.1. The mean and standard deviation of each activity

Activity	**Duration (Minutes)**	
	Mean	*Std. dev.*
Loading	2.49	0.23
Hauling	15.02	1.77
Dumping	0.83	0.18
Returning	15.55	2.76

3.5.3. Bayesian updating and DES model

The base simulation model was developed (Figure 3.7) using the input parameters shown in Table 3.1. It was assumed that total of 20,000 cubic yards of soil needed to be moved from the cut areas to the fill area. As data could be obtained from three trucks (out of five), it was also assumed that the project was run by only three trucks instead of five. The initial simulation predicted the project will be completed in 16 days. To match the project duration with the collected data, the activity durations were divided into 16 sub-sections, representing data from each day of the project.

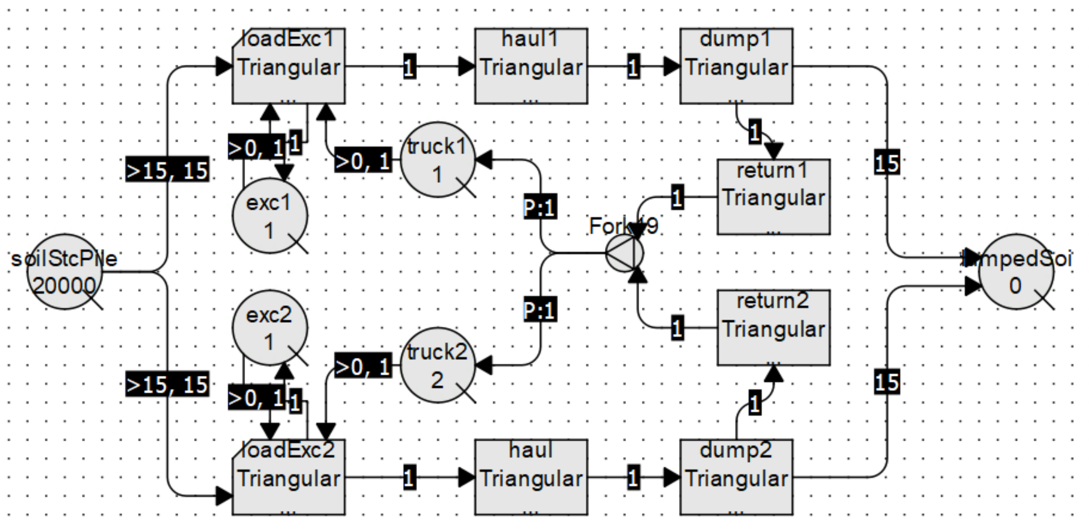


Figure 3.7. DES model of the earthmoving operations

It was decided to update the model every 2 days using the most recent durations. To update the input parameters (i.e., activity durations), the Bayesian technique was used to combine the initial estimates and actual sample data as described in the Methodology section. Figure 3.8 shows the comparison of prior and posterior distribution updated on day 2 for the loading time. We see that the prior was a normal distribution with a mean of 2.49 and a standard deviation of 0.23, which was updated to a posterior with a mean of 2.31 and a standard deviation of 0.19. This calculation was done for each of the four activities. So, the next prediction of the DES model was obtained using these updated input parameters. This process was continued every 2 days to get an updated prediction from the simulation model.

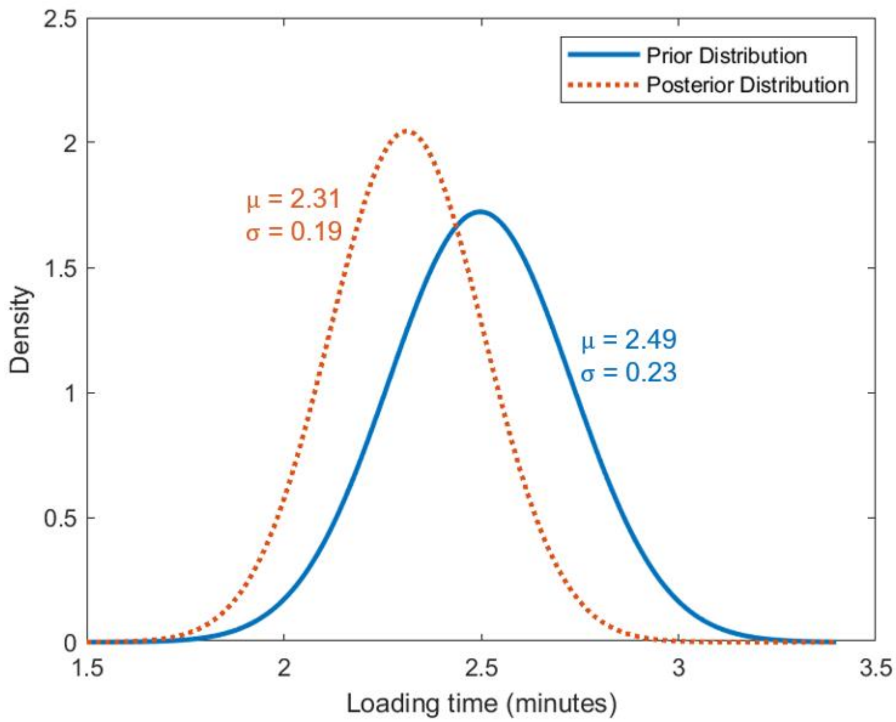


Figure 3.8. Prior and posterior normal distribution of the loading time after day 2

Figure 3.9 shows the percent completion of the project using a static prediction at the beginning (i.e., at day zero) of the project and dynamic prediction using the updated (at every 2 days) cycle times from the Bayesian update. The static simulation shows that the total project completion time was 16 days. However, when the cycle times were updated using the Bayesian updating technique and used as inputs in the simulation model, project completion time extended to 18 days. Figure 3.10 shows how cycle times changed throughout the project compared to the initial estimates. Following the updated cycle time, the productivity also changed, and we see how the productivity decreased with increased cycle time in Figure 10.

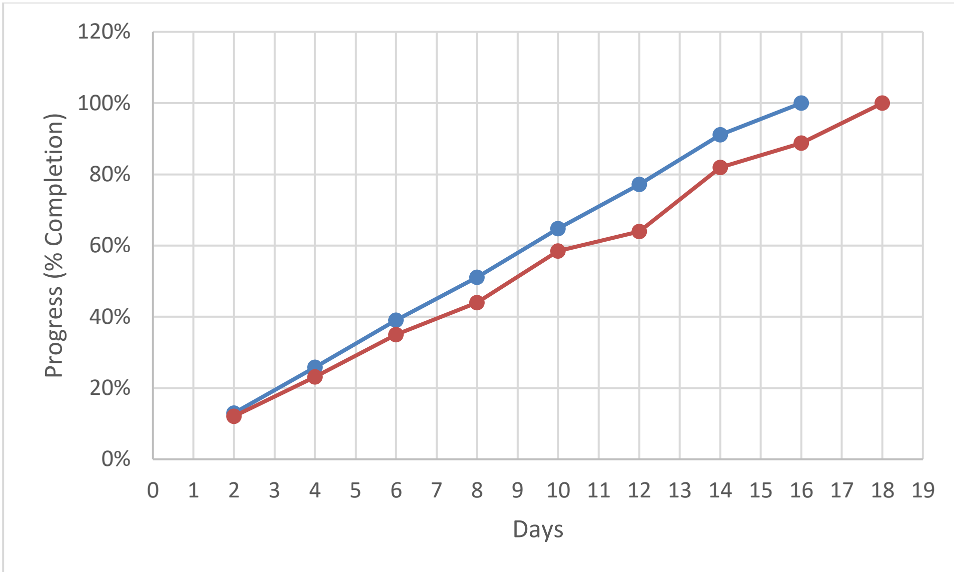


Figure 3.9. Percent completion of the project with initial vs. updated prediction

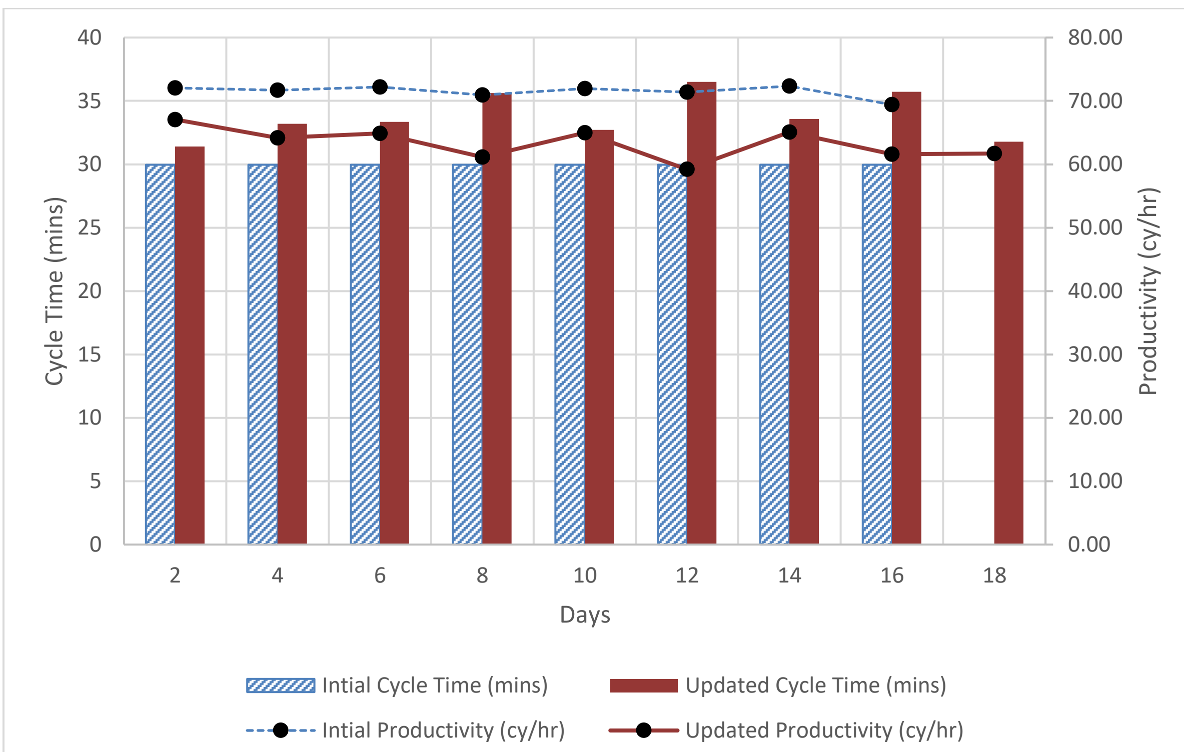


Figure 3.10. Cycle times and productivities using initial and updated cycle times

Figure 3.9 and Figure 3.10 show that simulations conducted during the construction might provide more accurate and reliable predictions as they used the actual data from the site compared to the initial prediction. These results show that the Bayesian updating technique was successfully used to update the distribution of the input parameters. Moreover, it leads to the conclusion that updating the simulation model frequently with actual data collected from the field can improve the prediction of the project performance by eliminating uncertainty in the original assumption.

3.6. Conclusions and future work

Earthmoving operations usually involve many unforeseen factors, such as constant change of the site layout, weather conditions, geological conditions. Thus, simulation can be a useful tool to experiment with multiple scenarios instead of resorting to costly experiments in the field. DES modeling is a powerful approach to simulate the earthmoving operation; however, the input parameters of the model are mainly obtained based on assumptions and previous experience, rather than actual data. This yields an inaccurate representation of the engineering process and leads to erroneous predictions for the project. To reduce the uncertainty of the model and improve simulation prediction, a proper updating technique is required. The updating technique can be also used to improve overall project control over schedule and cost. Towards this end, this chapter proposes an integrated approach to collect field data, identify activities of the equipment, develop a simulation model, and update the model using actual field data to provide an accurate prediction. A real-world earthmoving operation was chosen as a case study. Field data were collected using IMU attached to the excavators and dump trucks. Deep learning techniques, specifically, LSTM network was used to predict the activities of the equipment. Cycle times calculated from the predicted activities were used as inputs in the DES simulation model. Bayesian techniques were used to update the distribution of the input parameters of the simulation model. This demonstrates a formal approach to combine original estimates with the sample data collected from the site. The updated simulation prediction shows an increase in cycle times compared to the initial predictions as the project progresses. This shows reduced productivity, and the project duration was predicted to extend by 2 days compared to the initial prediction. These results can be used as a guideline to control the project cost and schedule.

This research contributes to the body of knowledge by demonstrating the capability of the DES model and machine learning algorithms to facilitate the decision-making process by simulating the most up-to-date field conditions. Moreover, the study contributes to the practice by presenting an integrated framework that contains an activity identification model for heavy civil equipment and a DES model simulating earthmoving operations. This provides a platform to automatically monitor the activities as well as estimate productivity by utilizing the most recent information.

The case study was performed only using data from the excavator and dump trucks. The future study can expand this work by including other earthmoving equipment (e.g., loader, scraper, grader, etc.) to capture the complex dynamics of typical heavy civil operations. Moreover, advanced algorithms such as reinforcement learning models can be integrated into the framework to automate the optimize process.

CHAPTER 4: AUTOMATED ACTIVITY IDENTIFICATION IN MODULAR CONSTRUCTION FACTORY

The content of Chapter 4 is an adapted version of the following manuscripts:

1. Rashid, K., Louis, J., (2020), "Activity identification in modular construction factory using audio signals and machine learning." Automation in Construction. 119, 103361. DOI: https://doi.org/10.1016/j.autcon.2020.103361

2. Rashid, K., Louis, J., "Automated active and idle time measurement in modular construction factory using inertial measurement unit and deep learning for dynamic simulation input". Accepted for Winter Simulation Conference 2021.

4.1. Introduction

Modular construction is an attractive and innovative project delivery method for buildings due to its advantages over traditional stick-built methods in terms of reduced waste, construction time and cost, control over resources and environment, and amenability to novel techniques in controlled factory settings. However, efficient and timely decision-making in modular factories requires spatiotemporal information about the resources regarding their locations and activities.

Per the overarching research methodology of this dissertation, the first step towards improving decision-making is the provision of automated means of data collection for monitoring construction progress in modular factories, which can then be input into decision-making methodologies such as simulation and optimization. This chapter describs the first part of this process– automated activity monitoring – using a ubiquitous data source present in every modular construction factory: sound. Audio data will be used to automatically identify commonly performed manual activities such as hammering, nailing, sawing, etc.

To develop a robust activity identification model, it is imperative to engineer the appropriate features of the data source (i.e., traits of the signal) that provides a compact, yet the descriptive representation of the parameterized audio signal based on the nature of the sound, which is very dependent on the application domain. In-depth analysis regarding appropriate features selection and engineering for audio-based activity identification in construction is missing from current research. Thus, this research extensively investigates the effects of various features extracted from four different domains related to audio signals (time-, time-frequency-, cepstral-, and wavelet-domains), in the overall performance of the activity identification model.

The effect of these features on activity identification performance was tested by collecting and analyzing audio data generated from manual activities at a modular construction factory. The collected audio signals were first balanced using time-series data augmentation techniques and then used to extract a 318-dimensional feature vector containing 18 different feature sets from the abovementioned four domains. Several sensitivity analyses were performed to optimize the feature space using a feature ranking technique (i.e., *Relief* algorithm), and the contribution of features in the top feature sets using a Support Vector Machine (SVM). Eventually, a final feature space was designed containing a 130-dimensional feature vector and 0.5-second window size yielding about 97% F-1 score for identifying different activities. The contributions of this study are two-fold: (1) A novel means of automated manual construction activity identification using audio signal is presented; and (2) Foundational knowledge on the selection and optimization of the feature space from four domains is provided for future work in this research field.

4.2. Background

4.2.1. Offsite and Construction

Off-site and modular construction is increasingly being seen as a promising solution to meet the growing housing demands in urban areas, due to its advantages over traditional on-site construction methods which include shorter construction through parallel work, reduced material wastage, and resulting cost savings for projects (Alazzaz and Whyte 2014). The highest degree of modularization is obtained in volumetric construction where modular units in the form of three-dimensional units for buildings are constructed off-site and minimal work is left to be accomplished on-site (Gibb and Isack 2003). A wide variety of built products are constructed in modular construction factories ranging from single-family homes to multi-family and office buildings, with a great variety and customization offered in terms of shapes and sizes amongst them.

The construction of these modular units in the factory involves different construction processes that are in separate workstations. Typically, the workstations of a modular construction factory can be divided into two categories: off-line stations and online stations. The off-line stations are typically dedicated to the creation of panelized components such as walls, floors, and ceilings from raw material. On the other hand, online stations are part of the assembly line for the volumetric unit, where various pre-made components (some from the off-line stations) are added and assembled to the modular unit. Therefore, the overall productivity of the factory depends on the productivity of both types of workstations. A delay at any of these workstations can potentially cause bottlenecks in production that can adversely affect the factory’s ability to meet the demand. Thus, to plan workstation layout and allocate resources to them, it is essential to know how much time is required to complete work at each station and to understand how that time is divided between various activities performed and idle times at the stations.

4.2.2. Automated productivity monitoring and control in modular construction

Several technologies have been proposed and adopted to ensure optimal performance of modular construction factories by focusing on the planning, monitoring, and control of modular processes. Researchers have integrated simulation models, RFID technologies, and optimization algorithms to develop control systems to assist operational decision-making. Mirdamadi et al. (2007) proposed a discrete event simulation (DES) and manufacturing executing system-based real-time production activity control framework for off-site construction.

Afifi et al. (2017) developed a combined approach using discrete-event and continuous simulation approaches to increase the productivity of modular construction. Azimi et al. (2011) illustrated an automated project monitoring and control framework using high-level architecture and radio frequency technology (RFID). Altaf et al. (2018) developed a production planning and control system using RFID, data mining, and simulation-based optimization in a panelized home production factory. Jureidini et al. (2016) developed a 3D/4D visualization tool to better perform different activities inside a modular construction factory. Moreover, lean tools and techniques have been implemented in modular construction facilities to explore their feasibility in reducing production time and waste (Moghadam 2014; Nahmens and Ikuma 2012; Yu et al. 2013).

The above research showcase efforts to integrate real-world data into planning models for improving modular construction. However, automated data collection efforts in modular construction have, to date, only involved the use of RFID tags to track the location of components at various stations. There is yet more research required to identify exactly what activities are being performed at the various stations themselves. This area has been explored in more depth in conventional onsite construction, which is described in the next section to identify potential solutions for application in the domain of off-site construction.

4.2.3. Audio-based classification

The broader field of audio-based classification can be defined as *machine hearing* (Lyon 2010), which is the quest for making computers automatically sense their environment using human-like acoustic sensing. Research efforts in this area aim to detect and classify different sounds such as speech, music, and environmental sounds, using their distinctive aural characteristics. For example, while music presents repeated stationary patterns such as melody and rhythm, speech signals have different traits in spectral distribution and phonetic structure. Environmental sound detection is a more complex domain due to the lack of periodicity, complexity in the spectrum, and an almost infinite range of phonemes and notes (Alías et al. 2016). Thus, depending on the domain of interest, audio-based classification takes various approaches for feature extraction, window selection, and learning model selection. For instance, window lengths between 10 to 50 milliseconds are typically selected to analyze and classify speech or transient noise events detection (Fu et al. 2011), whereas window length of several seconds is used in auditory scene analysis (Chu et al. 2009).

Similarly, there are many feature extraction approaches developed and implemented based on specific types of sounds and their applications. For example, speech, music, and environmental sounds generally present rich time-domain variation with diverse content, which can be parameterized by computing sign-change rate, signal power, etc. (Alías et al. 2016). Moreover, those dynamic variations can also be retrieved by a transformed domain, such as Fourier transform, cepstral-, or wavelet-domain (Haubrick and Ye 2019). In addition to retrieving physical properties from the audio, perceptual features can also be relevant and extracted from the audio input by modeling a simplified version of the auditory model of the human hearing system considering Mel, Bark, Gammatone filter-banks (Richard et al. 2013).

Audio signals have been explored to recognize daily human activities, such as walking, making coffee, brushing teeth, etc. in different built environments (García-Hernández et al. 2017; Liang and Thomaz 2019; Tremblay et al. 2015). Thus, it can be seen that there is a wide range of possibilities for analyzing an audio signal that varies depending on the domain of the audio being analyzed. The next section provides an overview of previous implementations of audio-based classification in construction.

4.2.4. Audio-based activity identification in construction

There have been recent applications of audio-based activity identification on construction sites. Sherafat et al. (Sherafat et al. 2019) proposed an automated activity recognition framework for construction equipment by fusing audio and kinematic signals. A binary support vector machine (SVM) model was trained using frequency-, and wavelet-domain features to identify value-adding and non-value-adding activities of a loader, a dozer, an excavator, a drilling machine, and a lift. In another similar effort, the audio was used to extract short-time Fourier transform (STFT) based time-frequency features and train an SVM model to identify productive and non-productive activities of the equipment (Cheng et al. 2017).

Sabillon et al. (Sabillon et al. 2020) proposed an audio-based Bayesian model for estimating cycle times of cyclic construction activities of equipment. A frequency-domain-based method, coupled with the hidden Markov model (HMM) was proposed to detect multi-layered construction activities from audio signals(Cho et al. 2017). Cheng et al. (Cheng et al. 2019) evaluated the software and hardware settings for audio-based activity analysis in construction operations by considering two different job site conditions (i.e., job site with a single machine and multiple machines), two types of SVM classifier (RBF and linear kernels), and two common frequency feature extraction techniques (short-time Fourier transform (STFT) and continuous wavelet transform (CWT)). Maccagno et al. (Maccagno et al. 2019) proposed a CNN-based approach for audio classification to identify what type of equipment was operating in the construction site. This study used Mel-spectrogram to extract Mel-scale features from the audio and demonstrated promising results in construction equipment activity classification.

Even though extensive research has been done aiming to improve production planning and control in panelized and modular construction factories, a detailed activity or task identification framework is missing from the literature. This is important as the activities performed at each workstation eventually contribute to the overall productivity of the factory. The nature of work (i.e., mostly using hand-held tools) inside a modular construction factory poses some practical challenges in implementing an active IMU-based framework while the unstructured movement of humans and materials poses obstacles toward implementing CV-based approaches. Moreover, existing audio-based activity analysis techniques for outdoor construction equipment may not be directly transferrable for manual activities in modular construction factories due to the difference in audio signals. Moreover, automated tracking of active and idle time in modular construction factories is missing from the literature.

4.3. Research gaps and point of departure

Based on the literature review performed, the following specific research gaps are identified and targeted in this chapter:

(1) Lack of robust activity identification framework for modular construction: Even though RTLS-, IMU-, and CV-based approaches exist in the current literature for on-site construction, an audio-based activity identification framework can address some of their limitations (e.g., unable to detect activities in stationary location for RTLS-based methods, attaching sensors to all the resources for IMU-based approach, and sensitivity towards light and occlusion for computer vision-based approach).

(2) Lack of audio-based activity recognition for manual activities: Even though audio-based activity identification frameworks for equipment in on-site construction exist in the current literature, temporal and spatial

characteristics of an audio signal generated from construction equipment are fundamentally different from the manual activities that are prevalent in building construction such as hammering, sawing, nailing, etc. Thus, there is a gap in knowledge regarding how sounds from manual activities from construction sites can be exploited towards extracting information about manual activities performed by factory workers.

(3) <u>Lack of systematic feature extraction method for audio-based activity identification in construction:</u> Compact yet descriptive feature extraction that is based on the type of audio signal and the targeted application is necessary to develop a successful activity identification framework. There is a clear gap in existing knowledge regarding an in-depth discussion on the process to extract, optimize, and synthesize features from different domains (i.e., time, frequency, wavelet, cepstral, etc.) to ensure successful implementation of a machine learning model for activity identification in the construction domain.

(4) <u>No system to track active and idle time manually:</u> There is no previous work on automated active and idle time identification for modular construction operations.

Considering the above gaps in knowledge, the goal of this chapter is to develop an automated activity identification framework to detect manual activities that occurs in the workstations of modular construction factories, as well as active and idle times in the workstations. This research goal is accomplished in this chapter through the pursuit of these specific research objectives:

(1) Develop and validate an audio-based activity identification framework for manual activities inside modular construction factories.

(2) Perform feature engineering for the audio-based framework which includes feature extraction from multiple domains (i.e., time, frequency, wavelet, cepstral, etc.), dimensionality reduction, and feature space design.

(3) Develop and validate a framework to automatically track active and idle times at different workstations in modular construction factories.

The scope of this research is to validate the proposed audio-based activity identification for the single tasks performed at a time. In practice, multiple activities are performed at the same time generating sounds simultaneously. However, due to a lack of prior literature in this area (i.e., audio classification for manual construction activities), this research seeks to first engineer appropriate features for this domain, which eventually can serve as foundational knowledge for future work in simultaneous multi-task identification.

4.4. Methodology

Towards accomplishing the stated research goal, this research developed an audio-based task identification framework, which is illustrated in Figure 4.1.

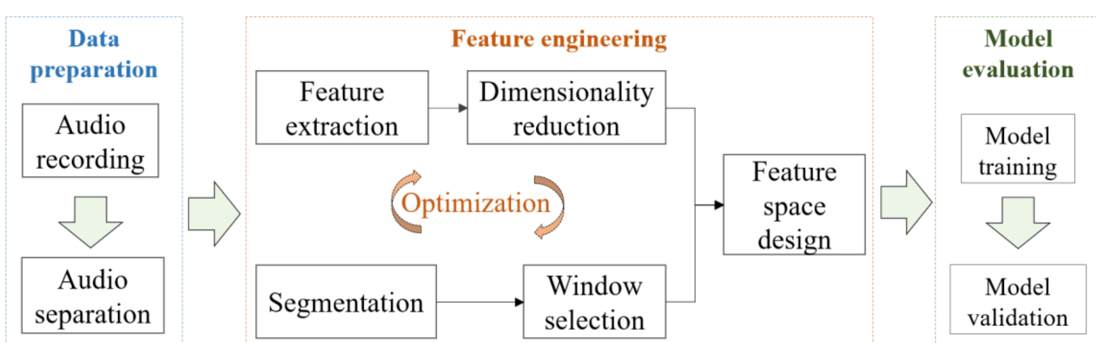


Figure 4.1. The overall framework for audio-based activity identification

The framework consists of three phases that include data preparation, feature engineering, and model evaluation. Major importance is given to feature engineering to develop a robust machine learning model. Previous studies in similar fields used features from different domains (time-, time-frequency-, cepstral-domain) separately, and did not evaluate how each of those features contributes to the overall performance of the model. This step of the methodology ensures that the final feature space is optimized, containing a compact and descriptive set of features. The collected data is first prepared for the feature engineering phase, wherein domain-specific features are extracted using different window sizes for feature space design. Four domain-specific features are tested in this study: time-, frequency-, cepstral-, and wavelet-domain. Finally, the optimized feature space and training dataset are used to train and evaluate the SVM model using the test dataset. The methodological steps to track active and idle time using IMU follow the same steps described in section 2.4. Thus, the following subsection will describe only the steps undertaken for the audio-based activity identification framework shown in Figure 4.1.

4.4.1. Data Preparation

The data preparation phase involves audio recording and separating audios of interest for this study. Data collection is performed by recording the sounds generated in a modular construction factory using commercial audio and video recorder. Audio associated with major activities (e.g. nailing using nail-gun, hammering, sawing using table saw, etc.) are then separated from the raw audio file. According to the scope of this study, multiple activities happening simultaneously are avoided to simplify the problem domain. Only single activities are considered and separated, and the emphasis is placed on designing the feature and windows. Thus, the separated audios are then used to engineer the feature space.

4.4.2. Feature Engineering

In this study, feature engineering refers to segmenting the input audio signal using various window sizes, extracting features from four different domains (i.e., time, frequency, wavelet, and cepstral), optimizing the features by using dimensionality reduction, and performing several sensitivity analyses. The following sub-sections discuss key components of the feature engineering process including (1) data segmentation, (2) feature extraction, and (3) feature and window optimization.

4.4.3. Data Segmentation

Data segmentation refers to the process of slicing the continuous time-series audio signal into discrete portions for feature extraction. In this study, two different levels of segmentation are used. The higher-level segmentation (referred to as "window") is used to segment the audio for feature extraction so that each window of audio produces a feature vector. For the second level: each window is further divided into discrete "frames" to calculate the short-time characteristics of the audio. Generally, the length of the frames is much smaller than the length of the window, and one window contains several frames. Different sizes of windows are used with 50% overlapping. Frame size and overlap are decided based on the type of features extracted from the corresponding frames. The features from each frame are used to extract statistical descriptors (i.e., mean, variance, etc.) or simply concatenated, depending on the type of features, to generate the feature vector for the corresponding windows. Figure 4.2 shows that the audio data are first segmented using W_m windows, and each window is then segmented using F_n frames.

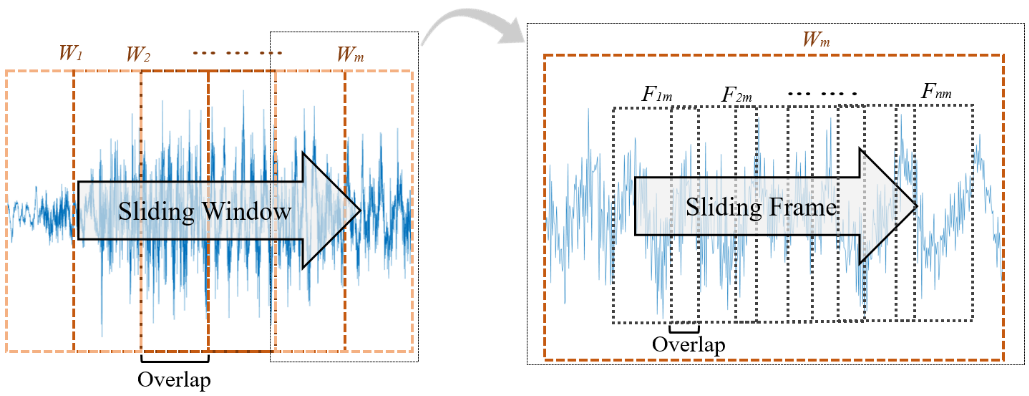


Figure 4.2. Segmenting the audio signal using windows and frames

4.4.4. Feature Extraction

Extracting compact and descriptive features from the audio is a key step to ensure successful machine learning applications for any classification problem. Feature extraction in audio classification is still an emerging field of research and based on reviewing state-of-the-art feature extraction techniques in audio classification in construction as well as in other research fields (e.g., speech, music, environmental sound classification), four domain-specific features are extracted in this study. They are time-, frequency-, cepstral-, and wavelet-domain features. Several features from each domain are extracted using the segmentation technique. Figure 3 shows the features obtained from the different feature spaces. These are described in greater detail in the following subsections.

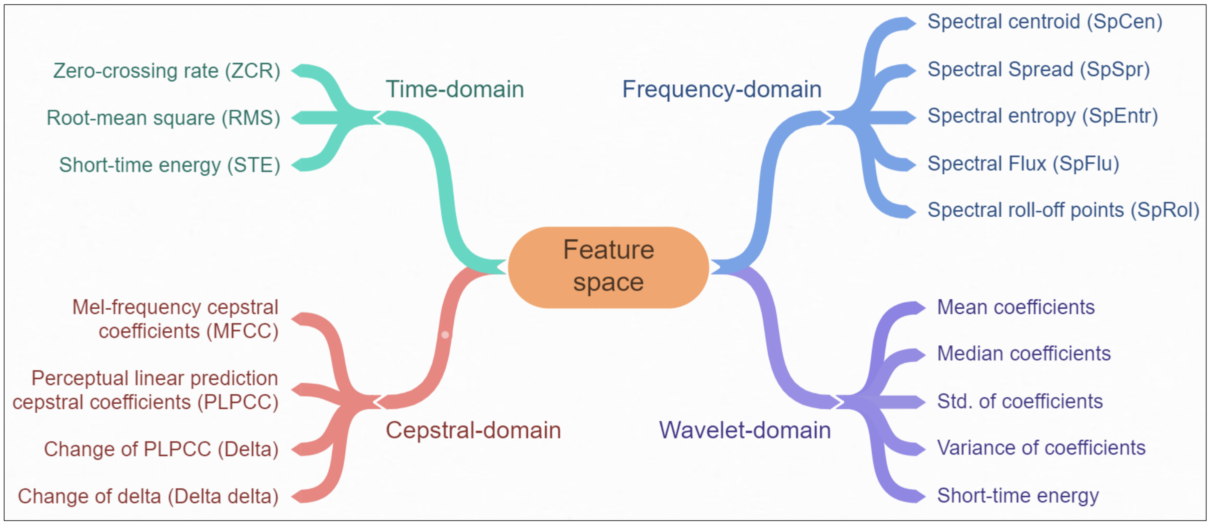


Figure 4.3. Summary of all features from four domains

4.4.4.1. <u>Time-domain features</u>

In the time domain, three types of features; zero-crossing rate (ZCR), root-mean-square (RMS), and short-time energy (STE) are extracted.

<u>Zero-Crossing Rate (ZCR):</u> ZCR is the rate of sign-change of a signal from positive to negative or from negative to positive. ZCR can be a rough representation of the frequency content and is calculated in short time windows. ZCR is calculated by the number of times the audio signal changed signs divided by the length of the frame. ZCR is calculated using the following equation and is illustrated in Figure 4.4.

If, $x_i(n) = 0,1, \dots, N-1$ is the sample of the i$^{\text{th}}$ frame.

$$Z(i) = \frac{1}{2N} \sum_{n=0}^{N-1} |\, sgn[x_i(n)] - sgn\, [x_i(n-1)]\, |$$

Z (i) is the ZCR for i$^{\text{th}}$ frame.

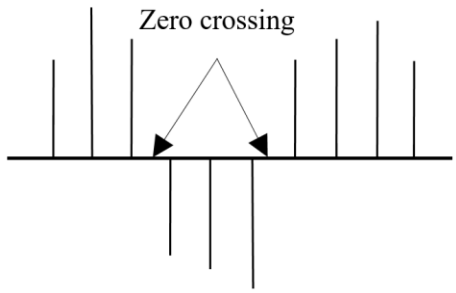


Figure 4.4. Illustration of zero crossing

Root-mean Square (RMS): RMS of an audio clip reveals the temporal variation of the signal's magnitude in terms of volume. RMS values are calculated on short time windows, like ZCR using the following equation.

$$sgn\ [x_i(n)] = \begin{cases} 1, & x_i(n) \geq 0 \\ -1, & x_i(n) < 0 \end{cases}$$

$$RMS(i) = \sqrt{\frac{1}{N}\sum_{n=0}^{N-1}[x_i(n)]^2}$$

The RMS values are divided by the frame length to remove any dependency.

Short-time energy (STE): Analogous to RMS, STE is also a simple representation of the energy in a signal and reveals how energy is varying through the time domain. The STE is calculated using the following equation.

$$STE(i) = \sum_{n=0}^{N-1}[x_i(n)]^2$$

Each of the time-domain features produces an array revealing the temporal variation of the audio signal. The arrays are concatenated to generate the feature vector for the corresponding window. Figure 4.5 shows the three time-domain features (i.e., ZCR, RMS, and STE) for a sample window. By looking at the waveform and the features, it can be noticed that all three time-domain features successfully capture the temporal variation of the signal.

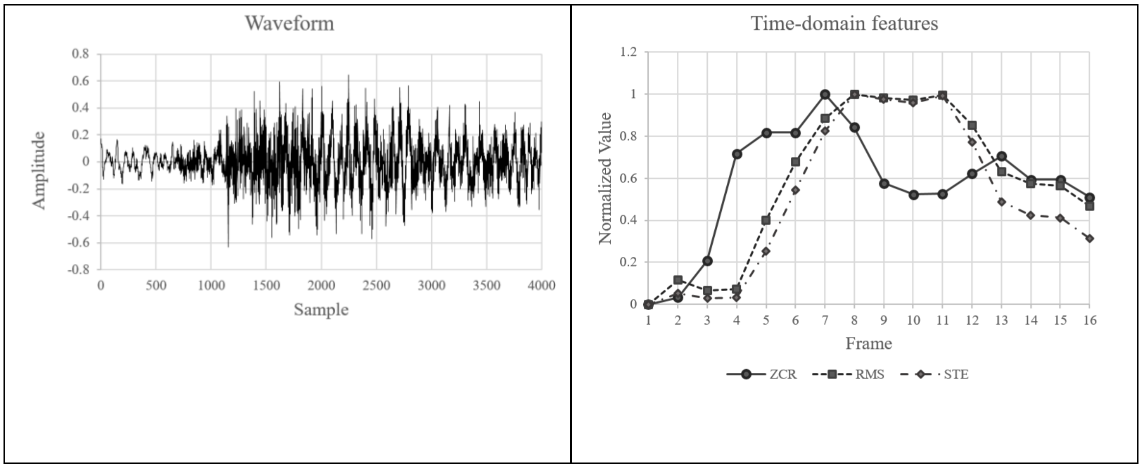


Figure 4.5. Three time-domain features with the audio signal of a sample window

4.4.4.2.Frequency-domain features

To extract frequency-domain features from the audio, first, a short-time Fourier transform (STFT) is applied to reveal time-localized frequency information from the signal. The STFT converts time-domain signals into a time-frequency domain, where the power of each frequency is illustrated by a color scale, and this time-frequency representation of the signal is also known as “spectrogram”. In this study, a hamming window of 512 sample size, 265 sample overlap is selected as a frame, and 512 FFT length is used during the STFT. There is a trade-off between time and frequency resolution in STFT depending on the size of the window. A narrow-band window produces better resolution in the time-domain, but the poor resolution in frequency-domain, and vice-versa. Thus, the window length of 512 samples is selected as a medium window. Figure 4.6 illustrates the process of implementing short-time Fourier transform (STFT) and generating spectrogram.

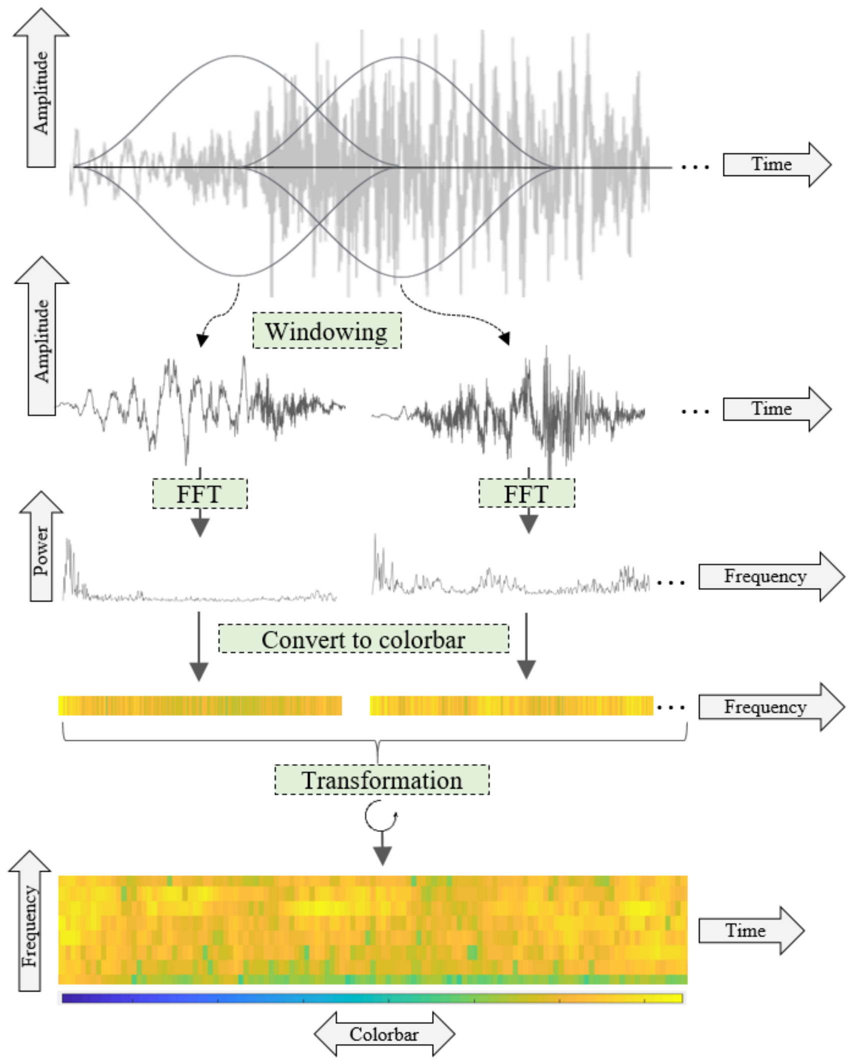


Figure 4.6. Steps of Short-time Fourier transform

Initially, the signal is segmented using a 512 sample size length into a finite number of frames. Fast Fourier transform (FFT) is applied to each of those frames to convert the signal from time-domain to frequency-domain. Finally, the intensity of the FFT can be illustrated using a color bar generating a time-frequency signal, a.k.a. spectrogram. From this time-frequency signal, several frequency-domain features, i.e., spectral centroid, spectral spread, spectral entropy, spectral crest, spectral flux, and spectral roll-off point are extracted. The spectral centroid can be defined as the center of gravity of the magnitude spectrum of the STFT. Spectral spread is the standard deviation around the spectral centroid. The spectral crest is the ratio of power spectral density (PSD) and the mean PSD. Spectral flux captures the sudden changes in the frequency energy distribution of the sound. And spectral roll-off is defined as the 95th percentile of the power spectral distribution and can be a measure of the skewness of the spectral shape.

Spectral Centroid: Spectral centroid is a representation of the "brightness" of the sound. This is measured by calculating the "center of gravity" of the magnitude spectrum of the STFT. Spectral centroid can be calculated using the following equation.

$$Spectral\ Centroid = \frac{\sum_{k=b1}^{b2} f_k * s_k}{\sum_{k=b1}^{b2} s_k}$$

Where f_k = frequency of frame k, s_k = spectral magnitude of frame k, and $b1$ and $b2$ are band edges of the frame k.

Spectral Spread: Spectral spread is the measure of the average spread of the magnitude spectrum of STFT in relation to the spectral centroid. This is a measure of the bandwidth of the sound. Spectral spread can be calculated using the following equation.

$$Spectral\ Spread = \sqrt{\frac{\sum_{k=b1}^{b2} (f_k - SC)^2 * s_k}{\sum_{k=b1}^{b2} s_k}}$$

Where SC = spectral centroid of frame k.

Spectral Entropy: Spectral entropy represents the "peakiness" of the STFT magnitude spectrum and is calculated by the following equation.

$$Spectral\ Entropy = \frac{\sum_{k=b1}^{b2} s_k log(s_k)}{\log(b2-b1)}$$

<u>Spectral Crest</u>: Similar to spectral entropy, the spectral crest is also a measure of the "peakiness" of the spectrum, where the higher crest specifies tonality and the lower crest indicates noise. The spectral crest is calculated by measuring the ratio of the maximum spectrum to the arithmetic mean of the spectral, as shown in the following equation.

$$Spectral\ Crest = \frac{max(S_{k[b1,b2]}}{\frac{1}{b2-b1}\sum_{k=b1}^{b2} s_k}$$

<u>Spectral Flux:</u> Spectral flux measures the variation of the magnitude of the spectrum over time. This feature is useful in detecting any onset in the audio. The following equation is used to calculate spectral flux.

$$Spectral\ Flux = \left(\sum_{k=b1}^{b2} |s_k(t) - s_k(t-1)|^p\right)^{\frac{1}{p}}$$

<u>Spectral roll-off point:</u> The spectral roll-off point is the bandwidth of the audio signal under which 95% of the total energy exist and can be calculated by the following equation.

$$Spectral\ roll - off\ point = i\ such\ that\ \sum_{k=b1}^{i}|s_k| = 0.95 * \sum_{k=b1}^{b2} s_k$$

All the feature arrays across all the frames are concatenated to generate a frequency-domain feature vector for the corresponding window. Figure 4.7 illustrates waveform, spectrogram, and all the frequency domain features of a sample window. The X-axis of the features is the frame number, and the y-axis is the normalized values. Figure 4.7 gives a visual snapshot of how the frequency-domain features are measuring the variation of the magnitude of the spectrum.

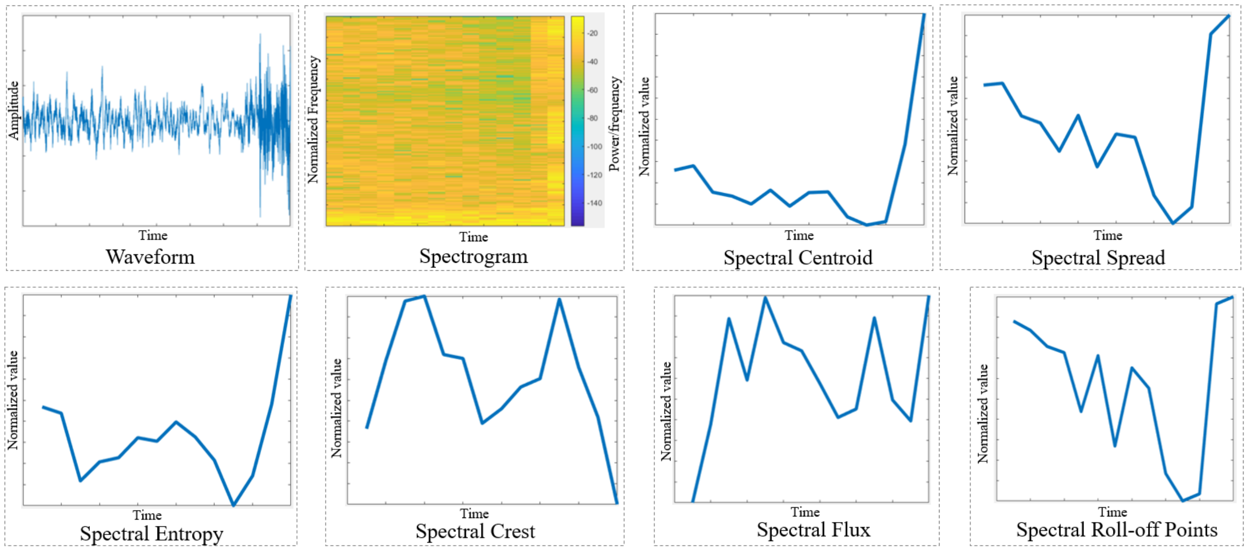


Figure 4.7. Waveform, spectrogram, and frequency-domain features of a sample window

4.4.4.3.<u>Cepstral-domain features</u>

Cepstral and cepstrum are derived by reversing the first four letters of spectral, or spectrum (Bogert 1963). A cepstrum reveals the information regarding the rate of change in a spectral band of audio. Cepstrum is derived by first taking a log of the STFT, and then again taking the spectrum of that log by discrete cosine transformation. In cepstral-domain, Mel-frequency cepstral coefficients (MFCC), perceptual linear prediction cepstral coefficients (PLPCC). During the feature extraction, a hamming window of 512 samples, and 256 sample overlap is used to segment each window of audio into several frames. The following sub-section discusses details on these two features.

<u>Mel-frequency cepstral coefficients (MFCC):</u> MFCC is a perceptual feature, where the frequency is converted from linear scale to a logarithmic scale, known as Mel scale. This is done to match the features more closely to human hearing as humans

are better at perceiving any changes in audio pitch at lower frequencies than they are at higher frequencies. The following equation is the formula to convert frequencies from linear scale to Mel scale.

$$M(f) = 1125 \ln(1 + f/700)$$

Where *f* is the frequency in the linear scale, and *M(f)* in the Mel scale. The steps to calculate the MFCC are shown in Figure 4.8.

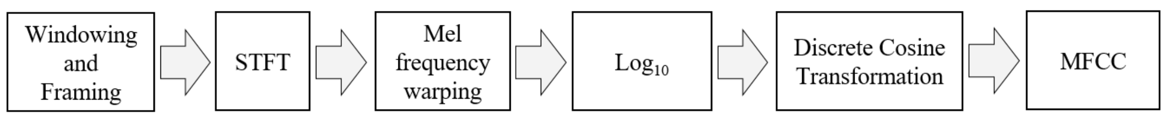


Figure 4.8. Steps involved in MFCC feature extraction

After windowing and framing the signal, first, the spectrogram is calculated using STFT. The magnitudes of STFT are then warped into Mel frequency using a Mel filterbank. The outputs from the Mel filterbank are then logarithmzed and transformed using discrete cosine transformation (DCT) to get the Mel-frequency cepstral coefficients. The first 13 MFCCs are used in this study. This means if there are *Fn* frames in one window, the feature dimension for MFCCs is 13X*Fn*. To reduce dimensionality, the mean, the median for each coefficient, as well as diagonal of the covariance matrix is selected as final features.

Perceptual linear prediction cepstral coefficients (PLPCC): PLPCC represents the spectral contour by using a linear prediction-based approach. Like MFCC, the linear frequency scale of the audio is converted to a different scale, known as Bark-scale, inspired by human hearing properties. Bark-scale is a psychoacoustic scale accounted for subjective measurement of the loudness of the audio and can be expressed by the following equation.

$$B(f) = 6\ln\left[\frac{f}{600} + \sqrt{\left(\frac{f}{600}\right)^2 + 1}\right]$$

Where *B(f)* is the frequency in Bark-scale, and *f* in linear scale. Figure 4.9 illustrates the steps necessary to calculate PLPCCs.

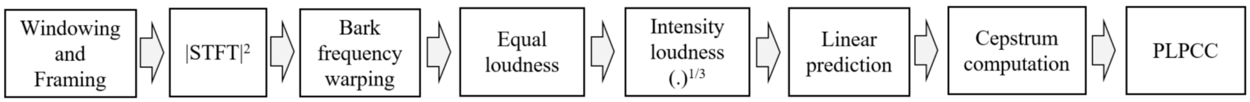


Figure 4.9. Steps required to compute PLPCCs

After windowing and framing the audio segment, the energy spectrum is computed using STFT. The magnitudes of STFT are then warped into Bark frequency using Bark filterbank. Then an equal-loudness pre-emphasis is applied to the filterbank outputs to simulate varying sensitivity of human hearing across different frequencies. Then a cubic-root amplitude compression is applied to represent the relation between intensity and loudness. The energy spectrum is then approximated using linear prediction. Finally, cepstral coefficients are extracted from the linear prediction.

12th order PLPCCs are calculated in this study, yielding 13 coefficients. This means if there are *Fn* frames in a window, the dimension of PLPCCs is 13X*Fn*. The mean, median, and diagonals of the covariance matrix are extracted to reduce the dimension of the feature space. In addition to PLPCCs, the changes of coefficients across frames (a.k.a. delta) and change of delta (a.k.a. delta delta) are also computed and used as features.

Figure 4.10 visualizes the transformation of frequencies from linear scale to Mel-, and Bark-scale. This figure contains a 10-second sample audio segment. First, a spectrogram is plotted using a 512 size hamming window and 50% overlap, which represents the original spectrogram. Then a reconstructed spectrogram is obtained from the MFCCs, titled "Reconstructed Spectrogram from Cepstra". The last plot shows the PLPCC spectra. It can be noticed that the power variation across the time is much more distinguishable when using MFCCs, and PLPCCs than linear frequency scale.

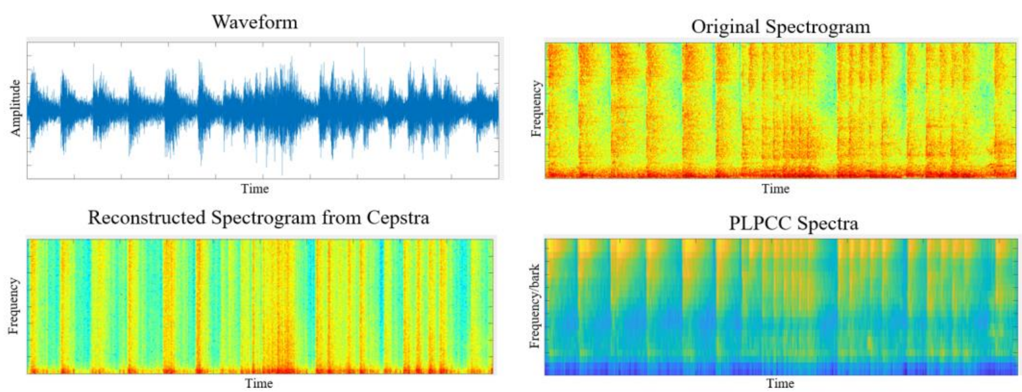


Figure 4.10. Transformation of a 10-sec audio segment into Mel- and Bark-scale

4.4.4.4. Wavelet-domain features

A wavelet is an oscillation with zero mean and finite length, represented by a mathematical function. One primary limitation of STFT is the theoretical limitation of Fourier Transform (FT) known as the uncertainty principle. This means that the smaller the size of the frame in STFT, the more information can be obtained regarding the location of the frequency in time, but less about the frequency value itself. On the other hand, wider frames in STFT reveals more information about the value of the frequency and less about the time. Thus, to analyze signals with dynamic frequency spectrum, a Wavelet Transform (WT) is frequently applied. Unlike FT, the WT has higher resolution both in the time and frequency domain. The basic differences of FT, STFT, and WT in terms of time, and frequency domain resolution is shown in Figure 4.11. The size of the blocks represents the resolution of the features in the time or frequency domain. The time series has high resolution in the time domain, and zero resolution in the frequency domain. On other hand, FT has high resolution in the frequency domain and zero resolution in the time domain. The STFT has medium resolution both in time- and frequency-domain and increasing resolution in one domain yields to decrease of resolution in another. However, WT has high-frequency resolution and low time resolution for smaller frequencies, and vice versa, high time resolution and low-frequency resolution for higher frequencies. These are particularly intriguing characteristics of WT for audio classification, as frequency-dependent features are important for lower frequencies, and time-dependent features are for higher frequencies, and WT makes this trade-off.

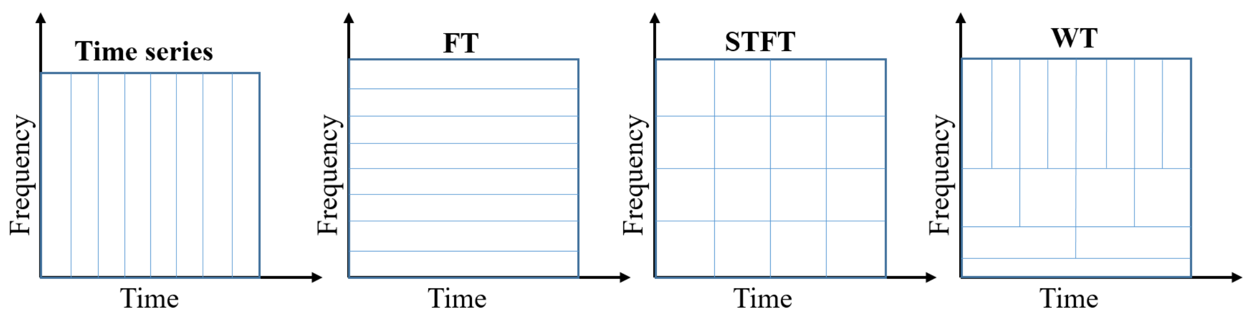


Figure 4.11. Schematic of time and frequency resolution of different transformation

Where sine and cosine waves are used in FT, which are infinite in length, wavelet has a finite length making it localized in time. Two major types of wavelets are discrete wavelet and continuous wavelet. The main difference between discrete and continuous wavelet is that scale and translation factors of discrete wavelet are discrete, where continuous wavelet can have an infinite number of scale and translation factors. This should be noted that discrete wavelets are not discrete in the time domain. Moreover, there are different wavelet families, with different shapes, smoothness, and compactness, and generally selected based on the specific application.

A discrete wavelet transform is used in this study. The process of a typical three-level DWT is illustrated in Figure 4.12. As seen from the figure, the signal is passed through a high-pass and low-pass filter. Outputs from the high-pass filter are called detail coefficients, and from the low-pass are called approximate coefficients. Both the coefficients are then down-sampled, generally to half of the length. Then approximate coefficients pass through the same process repeatedly. In this study, approximates coefficients from the last level (e.g., A3), and detail coefficients from all the levels (e.g., D1, D2, and D2) are used as features.

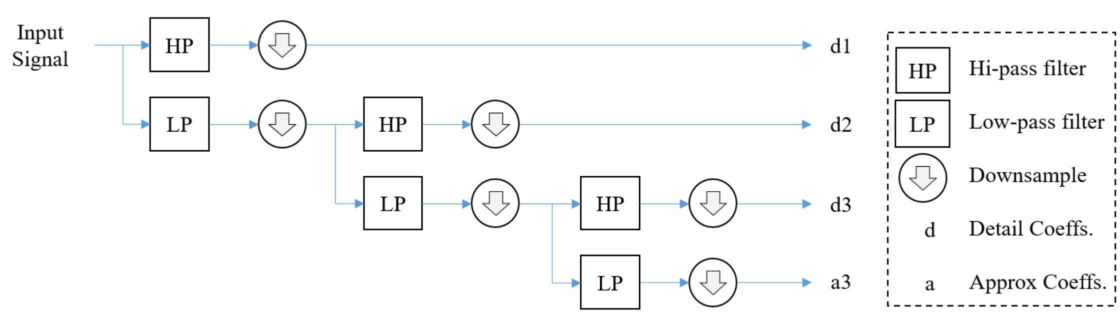


Figure 4.12. Process of Multi-level DWT

A *Symlets* wavelet family, specifically *'sym2'* with five-level decomposition is used in this study. Figure 4.13 shows a 250 ms (4000 sample) audio signal with the DWT parameters (without down-sampling), where *s* is the signal, a_5 is the level 5 approximate coefficients, and d_1-d_5 are detail coefficient at each level. The mean, median, standard deviation, variance, and short-time energy of each coefficient are selected for the final feature vector.

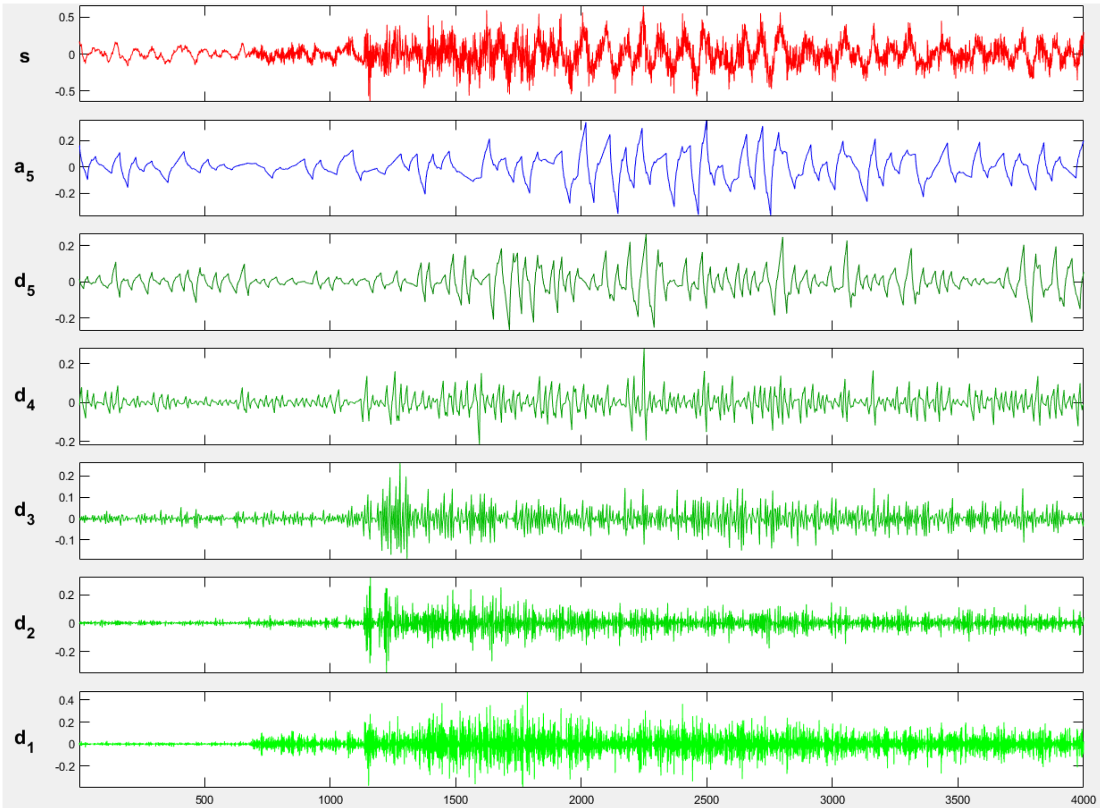


Figure 4.13. Five-level decomposition of a signal using 'sym2' wavelet

The next step is to identify the most important features from the feature space, which are representative of the audio, yet concise and computationally effective. The next sub-section discusses the feature selection approach taken in this study.

4.4.5. Feature and window optimization

One of the fundamental challenges of any classification problem is to accurately characterize the relationship between the features and the labels. However, only certain features in a dataset are descriptive and informative, and thus relevant to the problem domain. Irrelevant features are rarely differentiable, yet they contribute to the overall dimensionality and increase the complexity and computation time. In order to remove the irrelevant features from the data set, a Relief-based feature selection algorithm, known as ReliefF, is implemented. *ReliefF* calculates the feature weights or relevance to the labels using nearest neighbor concepts (Urbanowicz et al. 2018). The features are then ranked based on their respective weights. Different subsets from the features are used in several sensitivity analyses to examine their performance.

Since all the features are multi-dimensional and spread over different frames within a window, the window size is also an important factor in the sensitivity analyses. For example, a window containing *i* number of frames will generate $1\text{x}F_i$ dimensional features. Moreover, the size of the window should be small enough to provide high resolution, and wide enough to provide enough information to detect variation over time. Thus several window sizes, from 0.05 seconds to 1.5 seconds at different intervals are tested. Each sensitivity analysis containing different feature combinations, as well as different window sizes are validated and compared using support vector machine (SVM) and 5-fold cross-validation (CV).

4.4.6. Model evaluation

In this study, a multi-class support vector machine (SVM) is implemented in Matlab® for classifying activities. In multi-class SVM implementation, a one-vs-one setting is adopted, where each pair of classes is trained using one binary SVM with quadratic kernel function, and the final prediction decision is made using majority voting. 5-

fold cross-validation is used for parameter optimization, and 85/15 training and test ratio is used to test the final model. The main reason for choosing SVM as a classifier is to achieve the primary objective of this study, which is to compare various combinations of features from different domains, and window sizes to engineer the most optimum feature space. As training time from SVM is much shorter than the more recent deep learning architectures, quickly comparing different combinations of feature space becomes more computationally efficient. In this study, four performance measures; accuracy, precision, recall, and F-1 score are used to measure the performance of the SVM model. Moreover, a confusion matrix is used to examine the inter-class classification confusion among different activities. In addition to the overall precision, and recall of the model, those two measures are computed for each of the activities to see how they behave individually.

4.5. Case study

To validate the proposed methodology, field data were collected from a modular construction factory that contained several workstations dedicated to building floors and walls. Both video and audio data were collected using a commercial video camera. The audio was used for activity identification while the video was for labeling the audio data with associated activities. Video cameras were placed in two different locations for two consecutive days. Figure 4.14 shows the schematic diagram of the camera sources as well as the area of interest inside the modular construction factory. The data collection effort was focused on three specific workstations, one station with a table saw one partition wall (P-wall), and one long wall (L-wall) station as can be seen in Figure 4.14. Figure 4.15 shows snapshots from both camera sources.

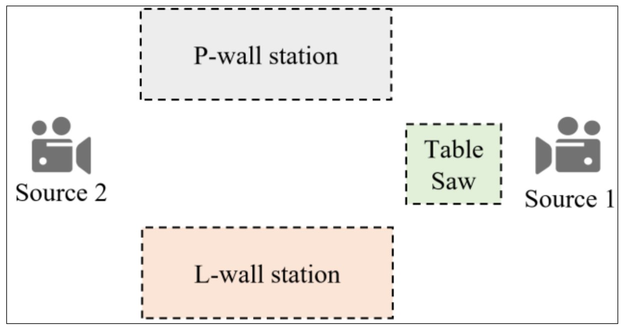


Figure 4.14. Schematic of the area of interest and camera sources

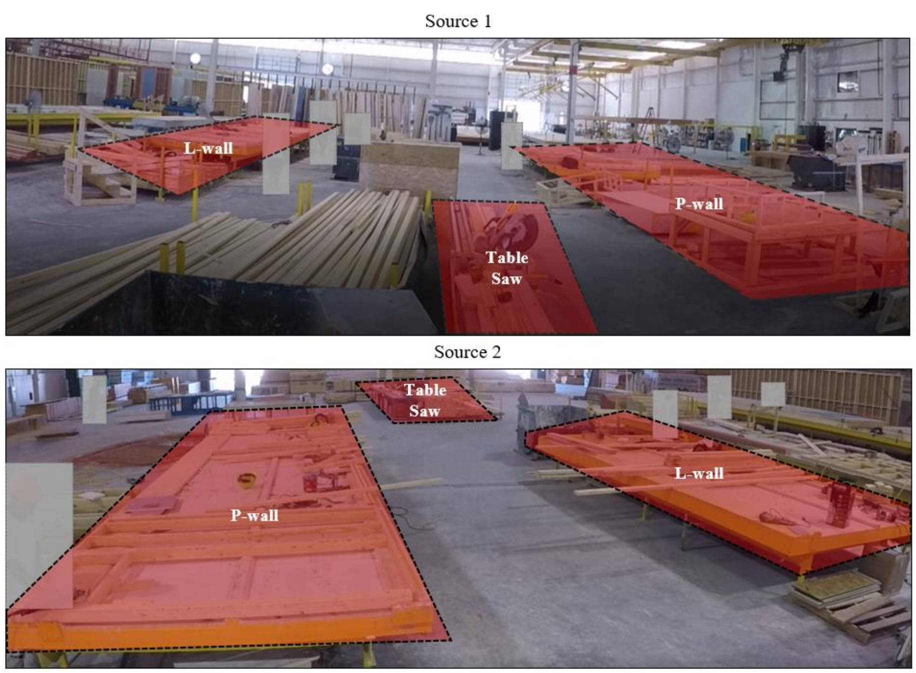


Figure 4.15. Snapshot from the video located at two different sources

Moreover, accelerometer data from both online and offline workstations were collected. For online workstations, six modular units were mounted with IMUs under the floor. For each unit, one full working day (i.e., 8 hours) data were collected. For offline workstations, three IMU units were attached under three tables where partial walls are built. Four hours of vibration data were collected for each of the offline stations. Figure 4.16 shows the IMU attachment positions in online and offline stations. The vibration data were collected with a sampling frequency of 10 Hz.

Figure 4.16. Location of the IMU attachment

4.5.1. Audio data collection and labeling

Audio from all other stations, except the highlighted stations in Figure 4.15 was ignored to simplify the data labeling process. From the captured audio, four major manual activities performed in those workstations were separated using the video as the reference and these include: nailing using nail-gun, hammering, cutting using table-saw, and drilling (denoted as NG, HM, TS, and DL respectively). In addition to the four activities, ambient sounds from the stations when they were idle were also separated, which is denoted as ID. Table 4.1 summarizes the different activities studied in this research.

Table 4.1. Different activities and abbreviations used in this research

Activity	**Abbreviation**
Nailing with nail-gun	NG
Hammering	HM

Sawing using table saw	TS
Drilling with electric drill	DL
No task/ Idle	ID

Figure 4.17 shows the waveform (i.e., time vs. amplitude) and spectrogram (i.e., time-frequency representation) of audio signals of the major activities. While some activities generate similar types of audio signals in the time-domain (first and third rows of pictures), there are variations present in the frequency-domain (second and fourth rows). For example, NG and HM produce almost identical waveforms, but NG has more power in higher frequencies than HM.

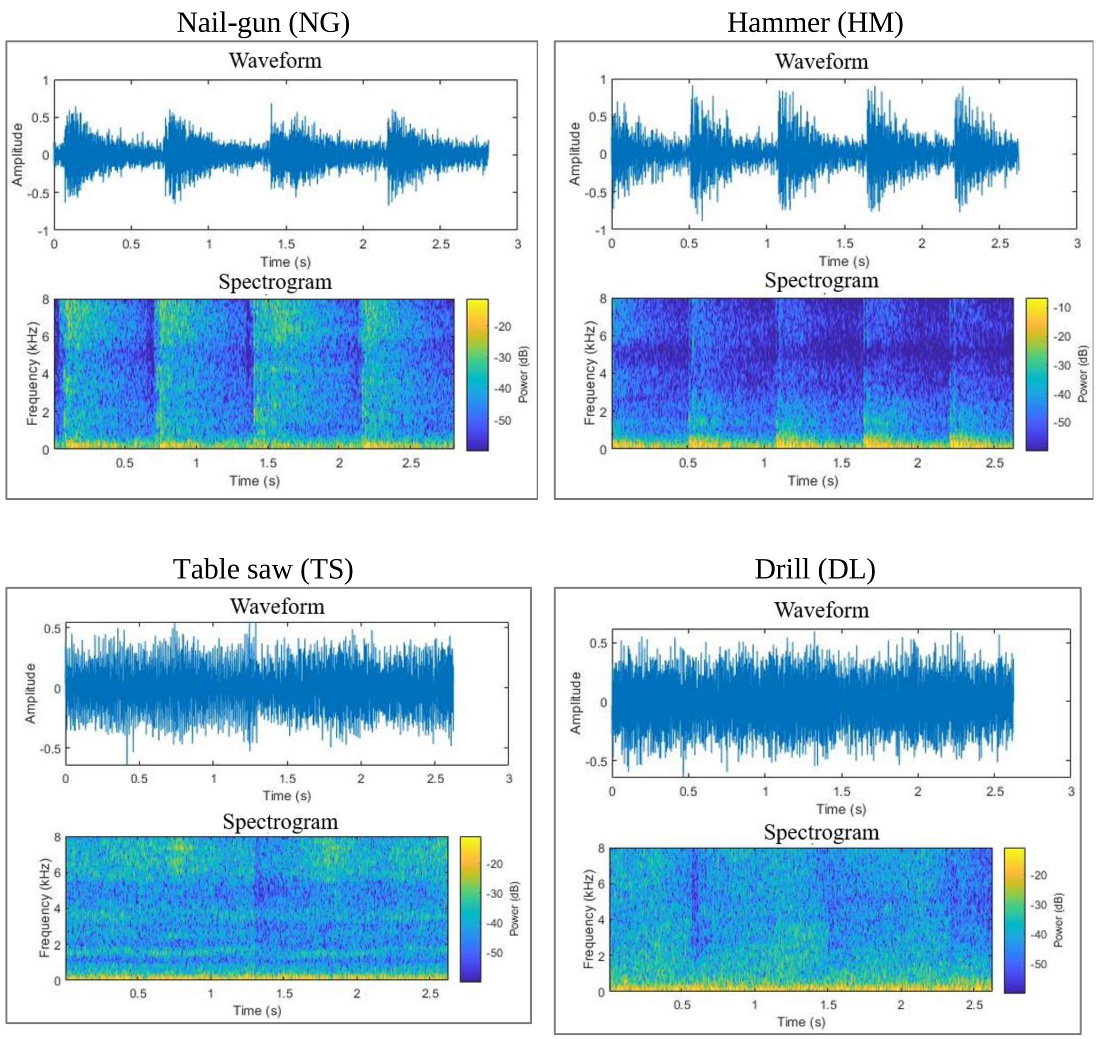


Figure 4.17. Waveform and spectrogram of four major activities performed in the factory

4.5.2. Audio data balancing

Figure 4.18 shows the proportion of data for each data label (task), and initially, DL and ID contained only 5% of the dataset where NG had 39%. This imbalance in the dataset can create a bias while training the classification model. This issue was addressed by adding white Gaussian noise to augment additional data for DL, ID, and HM activities. From Figure 4.18, after augmentation, each of the five activities contains a similar amount of data.

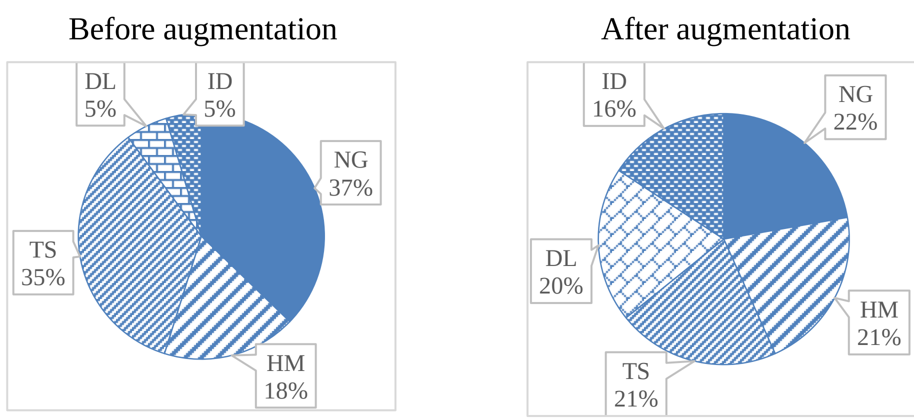


Figure 4.18. The proportion of data in each label before and after augmentation

4.5.3. IMU data collection and labeling

The dataset was then labeled using the reference videos and segmented into sequences using two different window sizes (15 seconds for online and 25 seconds for offline workstations), both with 50% overlap. However, as moving happens only when the modular units travel from one station to another, IMU data regarding *moving* activity are few compared to *active* or *idle*. This can create a bias in the training process. Thus, time-series data augmentation techniques presented in (Rashid and Louis 2019b) were used to augment *moving* data. Table 4.2 shows the data distribution in training, validation, and testing sets.

Table 4.2. Training, validation, and testing data distribution for online and offline stations

<table>
<tr><td rowspan="3">Class Labels</td><td colspan="6">Number of Sequences</td></tr>
<tr><td colspan="3">Online Stations
(15 sec. window size)</td><td colspan="3">Offline Stations
(25 sec. window size)</td></tr>
<tr><td>Training</td><td>Validation</td><td>Testing</td><td>Training</td><td>Validation</td><td>Testing</td></tr>
<tr><td>Active</td><td>12040</td><td>1720</td><td>3440</td><td>1376</td><td>197</td><td>393</td></tr>
<tr><td>Idle</td><td>20503</td><td>2929</td><td>5858</td><td>459</td><td>66</td><td>131</td></tr>
<tr><td>Moving</td><td>4715</td><td>674</td><td>1347</td><td colspan="3">N/A</td></tr>
</table>

Online workstations were labeled with three states: active, idle, and moving. As modular units pass through various online stations in the assembly line, it is important to know when a unit moves from one station to the next. This moving state will help to calculate the idle and active time at the different online stations. As offline workstations are stationary, only active, and idle states are labeled.

4.6. Results and Discussion

For the audio-based activity identification framework, data augmentation was applied to create a balanced training data set to eliminate any bias while training the model. Then parameter optimization was performed to find optimized features and window sizes. Upon selecting the parameter, the SVM model was trained and tested to examine the inter-class confusion of the model. For validating data balancing and parameter optimization, a 5-fold cross-validation approach was undertaken to compare before-after scenarios. For testing the final model trained with optimized parameters, 85%/15% of training and test data were used.

For the active and idle time prediction, the labeled data were split into training, validation, and testing datasets with 70%, 10%, and 20% ratios. Raw acceleration data were used as inputs to the LSTM layer with 100 hidden units. This layer mapped the input sequence into 100 features. The training and validation data set were used to train the model with fine-tuned hyperparameters.

4.6.1. Audio-based activity identification

Two SVM models were trained using data before and after augmentation to evaluate its effect. 0.25-second window size with 50% overlap, and time- and frequency-domain features were combined and used to train the SVM. 5-fold cross-validation (CV) yielded an overall accuracy of 72.3% before augmentation, and 75.4% after augmentation. However, only this improvement in overall model accuracy does not provide the complete picture regarding the performance improvement of the augmented classes. Thus, precision and recall values for each of the activities were calculated and provided in Table 4.3.

Table 4.3. Precision and recall of 5-fold CV before and after data augmentation

Activities	**Before Aug.**		**After Aug.**	
	Precision	Recall	Precision	Recall
NG	86%	84%	88%	83%
HM	77%	77%	77%	76%
TS	74%	90%	75%	92%
DL	25%	7%	50%	59%
ID	37%	17%	95%	93%

The table shows significant improvement in the precision and recalls for DL and ID after data augmentation. For example, the precision of DL was improved from 25% to 50% and recall from 7% to 59%. However, as only time- and frequency-domain features are used in this step, the overall performance of DL and ID is low. The next phase of the analysis examines the impact of features from different domains.

4.6.1.1. Feature Engineering

This step of the analysis investigates features from four different domains (time-, frequency-, cepstral-, and wavelet domain), optimizes them by fusing features from various domains, and eventually designs the optimal feature space using appropriate window sizes. Various window sizes from 0.05 seconds to 1.5 seconds were tested. This section is divided into two sub-sections: *dimensionality reduction* discussing feature optimization, and *window size selection* discussing the effect of different window sizes on the overall performance of the learning model.

The primary objective of dimensionality reduction is to reduce the dimension of the feature space by identifying the most descriptive (i.e., most contributing in distinguish the activities), and effective (computational efficient) features. Initially, a 318-dimensional feature vector was extracted from all four feature domains (48, 84, 156, and 30-dimensional feature vector from time-, time-frequency-, cepstral-, and wavelet-domain respectively). A 0.25-second window size with 50% overlap was used to train the SVM model and a 5-fold CV was used to evaluate the model using all combined features, as well as each domain-specific feature. Table 4.4 summarizes the performance measures (i.e., accuracy, precision, recall, and F-1 score) of SVM models based on domain-specific features, as well as combined features. Feature extraction time for each domain is also given in the “Time” column. The time is in millisecond per window. The “Dimension” column indicates the dimension of features from each domain. Figure 4.19 also visualizes the performance of different domain-specific features.

Table 4.4. Performance of SVM for features from each domain, their extraction times and dimensions

Feature Domain	Dimension of the feature vector	Accuracy	Precision	Recall	F-1 Score	Processing Time (ms/vector)
Time	48	71.5%	71.0%	75.0%	72.9%	3.7
Frequency	84	75.7%	68.0%	60.2%	63.9%	10.4
Cepstral	156	94.3%	92.0%	89.6%	90.8%	21.1
Wavelet	30	71.5%	70.0%	64.6%	67.2%	5.6
Combined	318	93.60%	91.00%	88.40%	89.7%	33.6

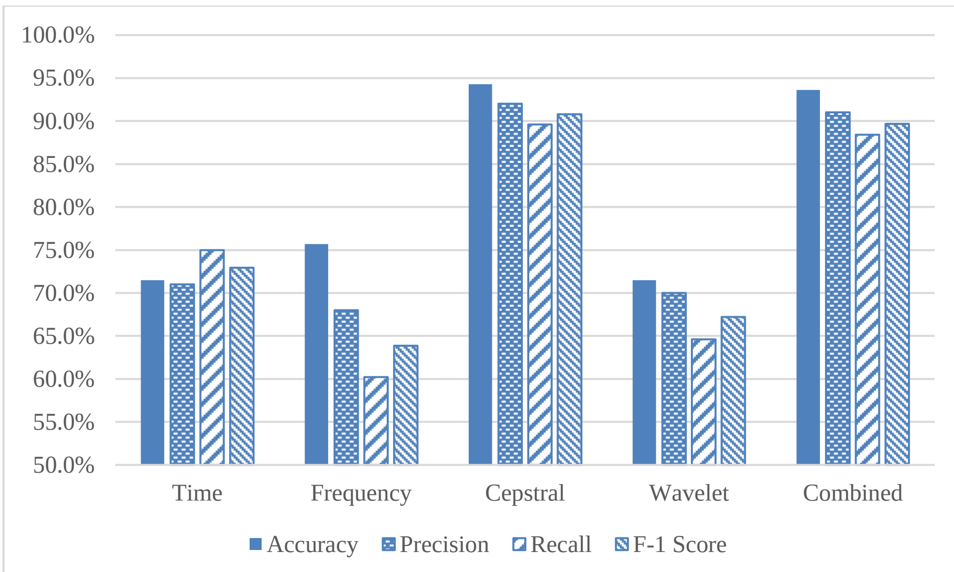


Figure 4.19. Performance measures of domain-specific features

Figure 4.19 shows that individually, cepstral-domain features demonstrated maximum CV accuracy 94.3%, precision 92%, recall 89.6%, and F-1 score of 90.8%. Cepstral-domain contains 156-dimensional features, and feature extraction time per window was 21.1 milliseconds. When features from all four domains were combined to a 318-dimension feature vector, the performance was lower than when only cepstral-domain features were used. This can be explained by the fact that not all features positively contribute to training the model. There might be some features that negatively contribute to the overall training process, creating more inter-class confusion, eventually reducing the performance of the model. Thus, it is important to investigate which features are representative, distinguishable, yet succinct in capturing the characteristics of the audio signals per specific activities. To do so, a feature selection algorithm, known as *ReliefF* was implemented.

The *ReliefF* algorithm calculated the importance weights of all the features based on the nearest neighbor approach and ranked the features based on their relative importance weights. Feature ranks and their weights are plotted in Figure 4.20. The rightmost portion of the figure shows that some features have negative weight, meaning they negatively contribute to the model's performance. A closer look at the crossing point (from positive to negative weight) of the plot reveals that the top 269 features are positive contributors, and the last 49 features act as negative contributors. Thus, in the next step, only positive 269 features were selected and several subsets of them were used to examine their impact on learning.

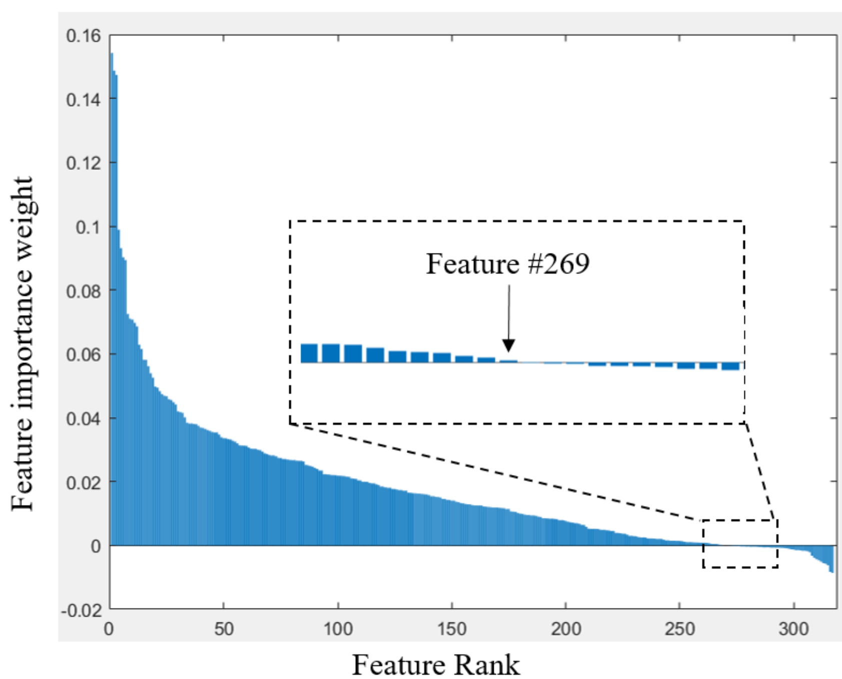


Figure 4.20. Results from the RelieF algorithm showing feature ranks and feature importance weights

Different subsets from all the positive features were selected by incrementing 10 features in each iteration (i.e., top 10 features, top 20 features, top 30 features, and so on), and SVM was trained for each of the subsets. The F-1 scores were computed using 5-fold cross-validation for each of the feature subsets and the result is plotted in Figure 4.21. A higher rate of improvement is noticed up to the top 170 features. After that, the performance of the SVM flattens while increasing the feature dimension. This implies that additional features after the top 170 only add extra dimensionality to the model instead of providing distinguishable information while classifying different activities. To reduce the dimensionality of the feature space, the top 170 features were further investigated in detail.

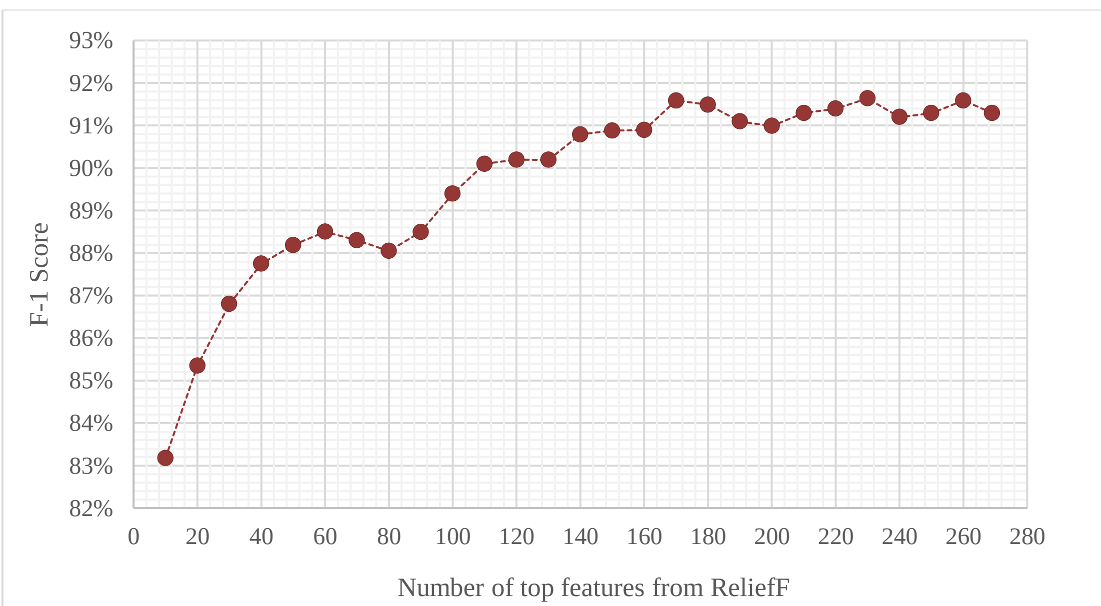


Figure 4.21. F-1 score for different subsets of the top features from *ReliefF*

Most of the features used in this study are multi-dimensional. Thus, it is more practical to look at which domain and feature class those top features belong to, rather than selecting them individually. For example, short-time energy (STE) is a $1 \times F_m$ vector, where *Fm* is the number of frames within a window. Some elements of STE could be within the top 170 ranks and some even might be negative features. Thus, it is important to examine how each feature and domain contributes to the overall feature space. To examine this, the number of elements of features from each domain was counted and their percentage contribution to the top 170 ranks was computed. The number of elements was normalized based on their length to remove size dependency. Figure 4.22 shows the percentage of domain-specific features present in the top 170 features retained from the *ReliefF* algorithm.

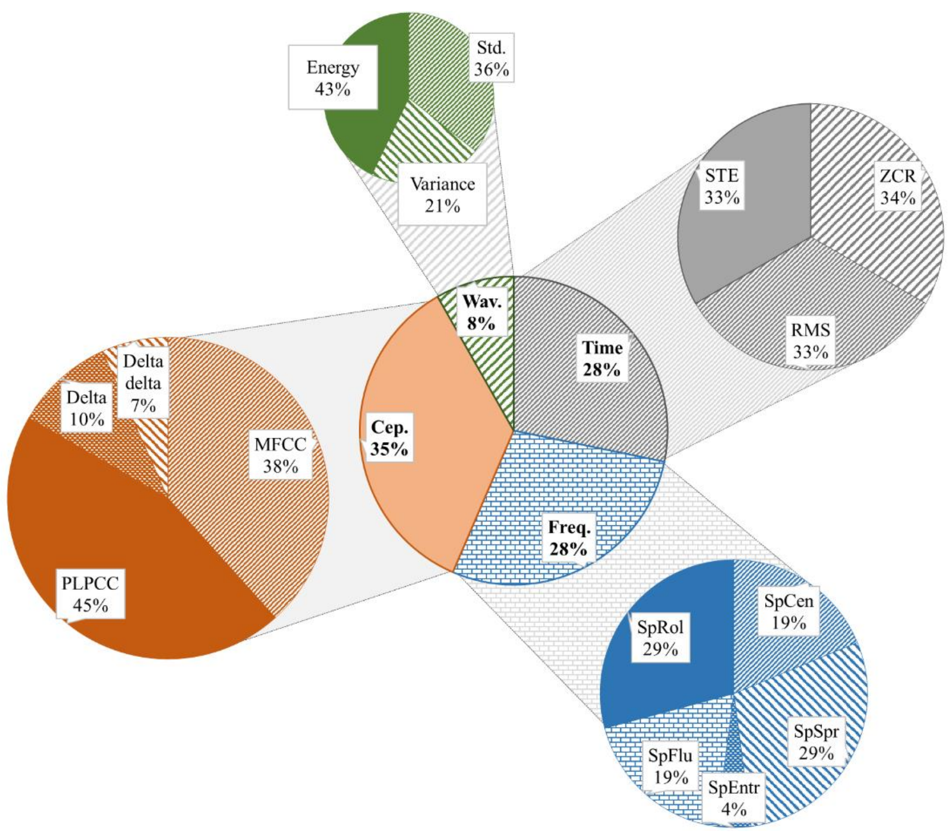


Figure 4.22. Contribution of features and associated domains in top 170 features from ReliefF

Figure 4.22 reveals that cepstral-domain features contribute the highest (i.e., 35%), and wavelet-domain features contribute the lowest (i.e., 8%) to the feature space. Each domain was also segmented to reveal the contribution of features within that domain. For example, within the cepstral-domain, MFCC and PLPCC contribute significantly higher (38% and 45% respectively) than delta, and delta delta (10% and 7% respectively). A final feature space with 130 dimensions was designed based on the contribution and summarized in Table 4.5. Wavelet-domain features were removed altogether from the final feature space due to their low significance. However, it will be misleading to conclude that wavelet-transformation is not an appropriate features extraction approach for audio classification in modular construction, as only one family of wavelets (i.e., Symlets) with one level of deconstruction (i.e., level 5) was used in this study. Other families of wavelets with different parameters can be explored to investigate the overall performance of wavelet transformation in audio-based task identification in the construction domain.

Table 4.5. Final feature space designed based on their contribution

Domain	**Feature**
Time	Zero-cross rate
	Root-mean-square
	Short-time energy
Frequency	Spectral centroid
	Spectral spread
	Spectral flux
	Spectral roll-off points
Cepstral	Mel-frequency cepstral coefficients
	Perceptual linear prediction cepstral coefficients

An SVM was trained using the final feature space using 0.25-second window length and 50% overlap, and the results of 5-fold CV are summarized as shown in Table 4.6. This table shows that the performance of the final feature space (90.9% F-1 score) is comparable with the top 170 features (91.6% F-1 score), even though the dimension is reduced from 170 to 130. This provides a practical means of designing the features which reduce the dimensionality (and thus complexity and duration) of the training process without unduly compromising performance.

Table 4.6. Performance of final feature space compared to top 170 features from *ReliefF*

Feature Space	**Accuracy**	**Precision**	**Recall**	**F-1 Score**
Top 170 features from *ReliefF*	94.6%	92.6%	90.6%	91.6%
Final 130 features based on contribution	94.1%	92.4%	89.6%	90.9%

Until this stage of the analysis, a fixed window size of 0.25 seconds and 50% overlap was used to segment the audio data. The following sub-section investigates different window sizes and their effect on the model.

4.6.1.2. Window size selection

Selecting an appropriate window size is an important step in any type of classification problem. A smaller window size provides good resolution but lacks details and vice versa. The window sizes selected in this study are 0.05 sec, 0.1 sec, 0.25 sec, 0.5 sec, 0.75 sec, 1 sec, 1.25 sec, and 1.5 sec. Precision, recall and F-1 score were computed for each window size and plotted in Figure 4.23. This figure demonstrates that the performance of the model increases initially with the increase of window length, however after a certain window length (i.e., 0.5 sec) the performance decreases. This demonstrates that a 0.5-sec window size provides a good resolution with enough details to classify different activities.

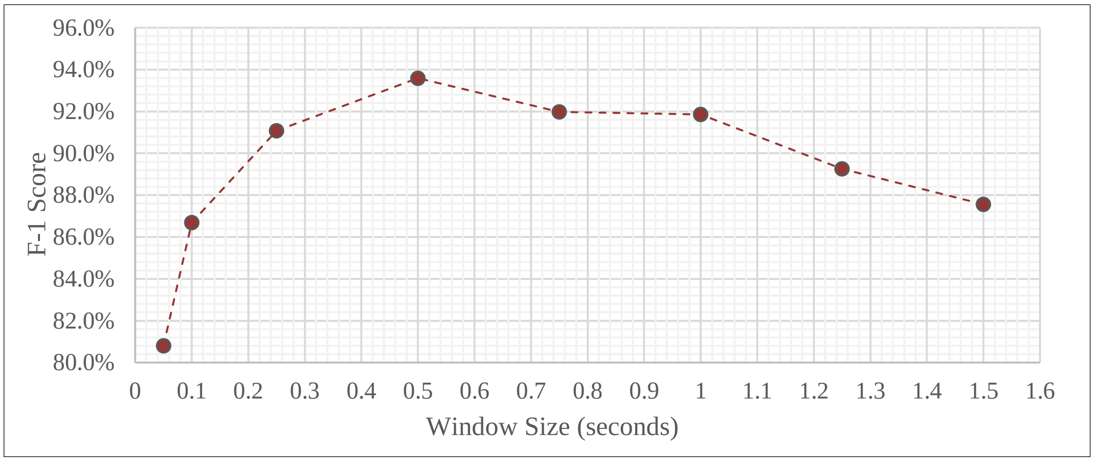


Figure 4.23. Cross-validation F-1 score for different window sizes

4.6.1.3. <u>Activity prediction</u>

In this section, the optimized parameters from the previous section were used to train an SVM model with the 130-dimension designed feature space (shown in Table 4) and 0.5-second window size with 50% overlap. Data were separated into training and test dataset using an 85%/15% train test ratio. Overall accuracy, precision, recall, and F-1 score are plotted in Figure 4.24. The trained model yielded to 96.6% F-1 score for test data.

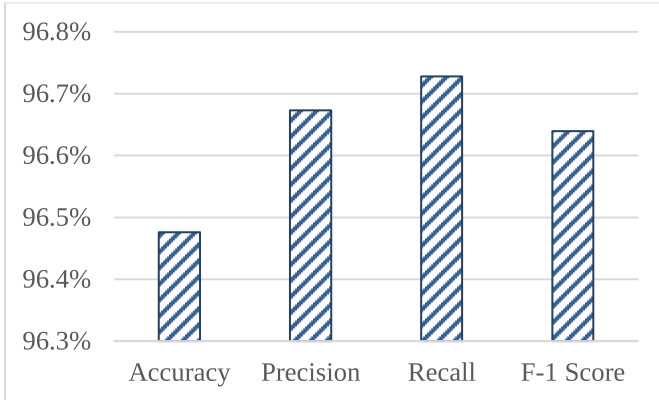


Figure 4.24. The test performance measures of the SVM model with optimized features and window size

Next, a detailed analysis was conducted to examine inter-class confusion of the SVM based on the test data set. A confusion matrix is plotted with the precision and recall of each activity in Figure 4.25. Figure 4.25 shows that ID demonstrates the highest performance with 100% precision and 100% accuracy. HM, NG, and TS demonstrate about 99%, 98%, and 97% precision and 94%, 97%, and 94% recall. Even though DL has a high recall value (about 99%), this class was confused with HM, and TS class about 10% of the time.

Actual Activity

Predicted Activity	DL	HM	ID	NG	TS	Precision
DL	90	5	0	0	5	90.00%
HM	1	90	0	0	0	98.90%
ID	0	0	71	0	0	100%
NG	0	1	0	98	1	98%
TS	0	0	0	3	89	96.70%
Recall	98.90%	93.80%	100%	97.00%	93.70%	

Figure 4.25. Confusion matrix of the SVM

Finally, a 25-second segment was separated from the raw audio, and the activities labels were predicted using the model trained in the previous step (i.e., with optimized parameters). The actual activities and predicted activities are shown in a stair-step graph in Figure 4.26. The figure shows that the model could predict correct activities with occasional misclassification.

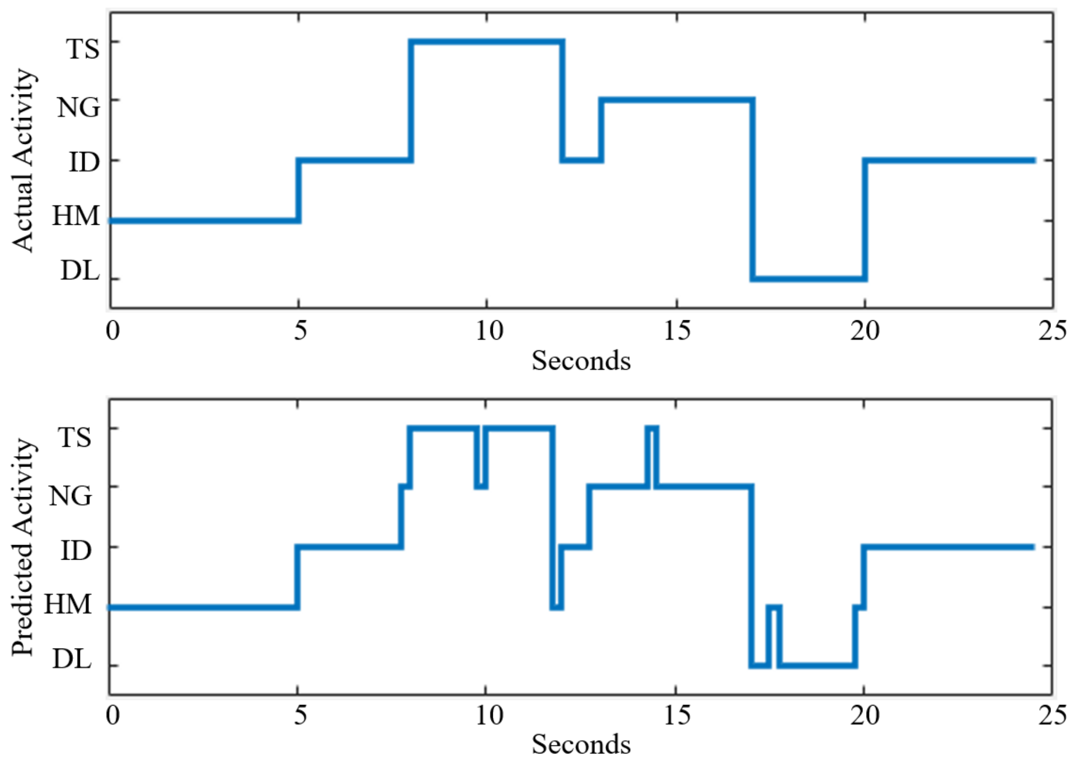
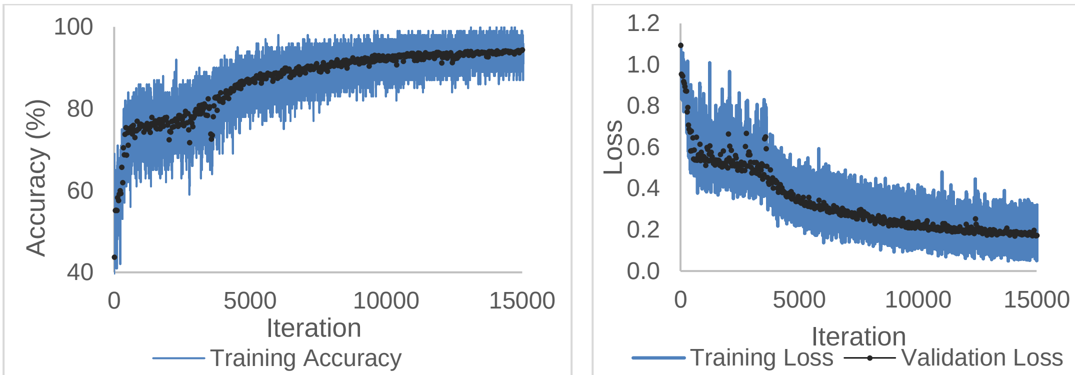


Figure 4.26. Actual and predicted activities of 25-sec audio segments

4.6.2. Active and idle time tracking

An LSTM network was trained to track the active and idle time in the workstations. The labeled data were split into training, validation, and testing datasets with 70%, 10%, and 20% ratios. Raw acceleration data were used as inputs to the LSTM layer with 100 hidden units. This layer mapped the input sequence into 100 features. The training and validation data set were used to train the model with fine-tuned hyperparameters. Figure 4.27 shows the training progress of the LSTM network of the online workstations.



Training and validation accuracy Training and validation loss

Figure 4.27. Training progress of the LSTM network for online workstations

The validation accuracy for online stations was 94.1% and for offline stations was 94.8%. After training the LSTM, the testing data was used to evaluate the performance of the model. Table 4.7 shows the evaluation results of the model tested with the testing data.

Table 4.7. Accuracy, precision, recall, and F-1 score of the LSTM model for online and offline stations

	Accuracy	Precision	Recall	F-1 Score
Online Stations	93.4%	93.4%	93.8%	93.6%
Offline Stations	93.5%	92.1%	90.9%	91.5%

The trained LSTM models demonstrated a 93.6% F-1 score for online stations and a 91.5% F-1 score for offline stations. Even though the F-1 score represents the overall performance of the network, they do not provide information regarding the misclassification of different classes. Thus, a confusion matrix was used to identify the classes that are misclassified as shown in Figure 4.28.

Prediction \ Ground Truth	Active	Inactive	Moving	Precision
Active	3026	240	48	91.3%
Inactive	363	5616	0	93.9%
Moving	51	2	1299	96.1%
Recall	88.8%	95.9%	96.4%	

Prediction \ Ground Truth	Active	Inactive	Precision
Active	373	14	96.4%
Inactive	20	117	85.4%
Recall	94.8%	89.3%	

Online workstations Offline workstations

Figure 4.28. Confusion matrix of LSTM network

The confusion matrix of online stations shows that *Moving* demonstrates the highest precision and recall. Moreover, *Moving* is mostly misclassified with *Active*, which is understandable as some *Activities* may have a very similar signal pattern as *Moving*.

Next, one random section was selected from the dataset for both online and offline workstations to calculate active and idle time from the trained model. An 8.5-hour dataset for the offline station was used as input to the trained model. Figure 4.29 shows the ground truth and the prediction for the online stations. We can see there were two *Moving* instances in the ground truth, where five were predicted by the model. A closer look reveals that each of the three misclassified *Moving* instances occurred during the *Active* class. The original dataset contained very few *Moving* classes, as, after a couple of hours of activities, the modular units are moved to the next station using an electric pusher, which takes about 15 to 30 seconds. To balance the dataset, augmentation techniques were used to generate synthetic *Moving* data. A similar plot was drawn for the offline stations in Figure 4.30. Figure 4.30 contains only the *Idle* and *Active* state of the offline workstations.

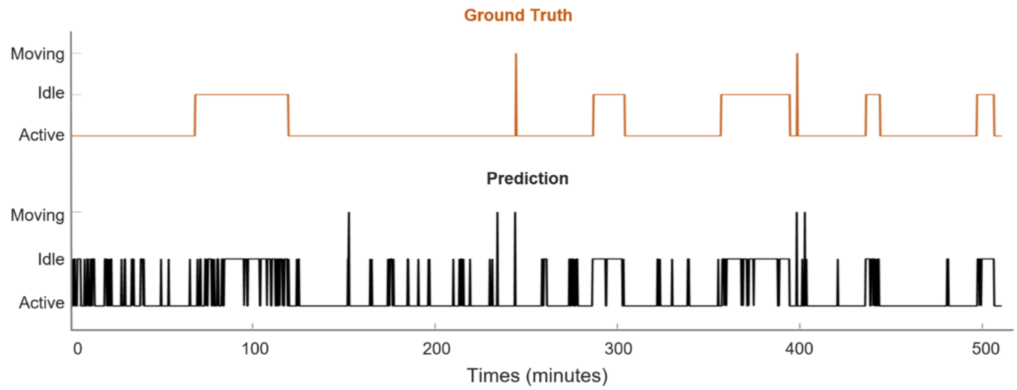


Figure 4.29. Ground truth and prediction of the LSTM model for online stations

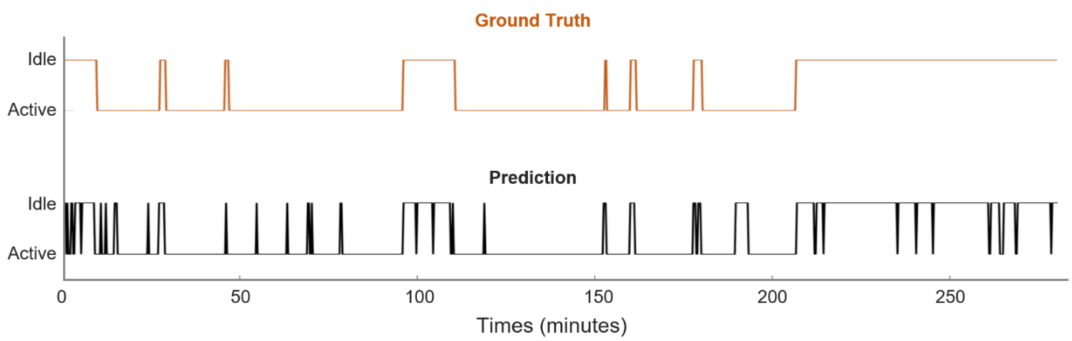


Figure 4.30. Ground truth and prediction of the LSTM model for offline stations

Finally, idle, and active time was calculated using the predictions shown in Table 4.8. We see that, for offline stations, out of the 8.5 hours 387.2 minutes were active, and the prediction was 380 minutes with a 1.8% error. Similarly, the online workstation showed a 1.4% error in calculating active time from the prediction.

Table 4.8. Active and idle time calculation from the trained LSTM model

	Time (minutes)			
	Offline Stations *(8.5 hours)*		**Online Stations** *(4.6 hours)*	
	Active	Idle	Active	Idle
Ground Truth	387.20	123.30	175.00	105.88

Prediction	380.00	129.70	177.50	103.33
Error	1.80%	5.19%	1.40%	2.40%

4.6.3. Summary of results

The segmented audio data was first balanced using data augmentation to ensure unbiased training. Significant improvement was noticed after data augmentation, especially for underrepresented activity, *drilling* in the dataset. Next, four domain-specific features were analyzed to find and select the most appropriate features which are representative, distinguishable, and concise. In doing so, several sensitivity analyses were conducted, and a final feature space with 170 features was designed.

The result of the feature engineering demonstrated that the cepstral-domain features contribute most (35%) among the four domains in the top 170 features. PLPCC and MFCC were found to be the two most important features, contributing about 45% and 38% respectively in the cepstral domain. This explains the fact that the features that are inspired by human hearing properties tend to contain most spatiotemporal information of the audio signal. Both MFCC and PLPCC are perceptual features where the signal is converted to different scales (i.e., Mel-scale for MFCC, and Bark-scale for PLPCC) to match the features more closely to human hearing.

Next, different window sizes were tested to select an appropriate window size, which gives higher resolution as well as contains enough information to classify activities accurately. By doing so, a 0.5-second window size was selected as it yielded to maximum F-1 score for the model. This can be explained by observing the waveforms of the four selected activities in Figure 7. While TS (i.e., table saw) and DL (i.e., drilling) audio signals maintain a roughly constant power throughout the temporal domain, NG (i.e., nail gun) and HM (i.e., hammering) signals have a variation of power with time and the variation can be roughly wrapped with 0.5-seconds window. This explains the reason why a 0.5-second window size yielded to maximum F-1 score for the SVM model.

The data augmentation, and parameter (i.e., feature, and window size) optimization was validated using 5-fold cross-validation. After optimizing the parameters, an SVM model was trained and evaluated using the 85%/15% training test dataset ratio. The test result demonstrated a 96.6% F-1 score. A confusion matrix with precisions and recalls was plotted to examine inter-class confusion. The highest misclassification was observed for the DL class, which was confusion with NG class 5% of the time, and with TS class other 5% of the time. This may be because the audio signal from intermittent drilling showed some similarity with the nailing activity, while continuous drilling has some similarity with the sawing activity. Finally, a 25-second segment from the raw audio was separated and predicted using the trained SVM with optimized parameters to demonstrate how a real-world application could look like.

For the active and idle time tracking, this study also investigated the potential of vibration generated from the activities performed to identify the active and idle time using a deep learning approach. One of the major challenges was to choose the appropriate window size as an *active* sequence can have multiple *idle* durations in between and considering those as idle time is not purely logical. Thus, a larger window size (i.e., 15 seconds for online stations and 25 seconds for offline stations) was considered in this study. The trained LSTM network demonstrated a 93.6% F-1 score for online stations and a 91.5% F-1 score for offline stations. The prediction of the LSTM network showed the capability of automatically measuring active and idle time with an average error of 2.7%.

4.7. Major findings and research contribution

The proposed framework in this chapter consisted of two major components: an audio-based activity identification framework, and an IMU-based active and idle time tracking for modular construction operations

The major findings of this study are:

- Similar to the findings of Chapter 2 (i.e., for IMU-based activity identification), deep learning network and data augmentation significantly improves the performance of the audio-based activity identification model.
- For audio-based classification, it is essential to engineer features from various domains (e.g., time-, frequency-, cepstral-, and wavelet) to maximize the performance of the model.
- Vibration generated at the workstations while working can be used to reliably track active and idle time in a modular construction factory.

The specific contribution of this chapter to the body of knowledge and practice are:

- This study demonstrates the potential of using environmental audio to automatically identify manual activities inside a modular construction factory.

- Audio data augmentation was found to improve the performance of the activity identification model and can reduce the need for training data for such methods.
- The systematic approach presented in this chapter to engineer appropriate features can be applied to identify any type of construction worker from the audio signal.
- Active and idle times can be automatically and accurately tracked using the vibration generated from work performed in the workstations, which can lead to automatic productivity monitoring.

4.8. Conclusion

Identifying, tracking, and monitoring activities in a modular construction factory is a key step in productivity assessment. Towards this end, this chapter proposed and validated an automated audio-based manual activity identification framework for modular construction. Also, an IMU-based framework was proposed which successfully tracked idle and active time in different workstations. The proposed methods can be applied to modular construction factories, possibly in combination with RTLS-, or CV-based activity recognition approaches for automated data collection (e.g., cycle time, value-adding time, active/busy time, etc.), as well as real-time monitoring of the factories.

This information eventually can support data-driven decision-making, possibly coupling with dynamic simulation models. Even though the proposed method was validated for a modular construction factory, this framework can be extended for on-site constructions as well, where many activities are performed manually (e.g., stick-built house building). In developing the framework particular focus was given towards engineering appropriate features, which essentially is a paramount step in ensuring robust model training and validation. The methodology, especially how the feature space was engineered in this study can serve as a platform for future endeavors in this domain and can be extended to other types of activities with deep classification architecture (e.g., convolutional neural network, recurrent neural network, etc.).

4.9. Limitations and Future Work

The scope of this research was to identify single activities happening at any point in time. However, in real-world conditions, multiple activities happen simultaneously, thereby generating a layered audio file. This problem falls under the multi-label time series classification domain. Future work will use the results obtained from this study, especially regarding feature engineering, to perform simultaneous activity identification using deep learning approaches. Source identification from different workstations using a single microphone is another challenge in audio-based approaches. Towards this end, beamforming techniques using microphone arrays can be adopted to differentiate and separate audio sources based on their location of origin to analyze station-based activities.

Another challenge of the audio-based approach as a stand-alone application is to identify activities that are value-adding, but which do not generate distinct sounds. This problem can be addressed by adding another layer of pattern recognition model on top of the classification model or using windowing techniques to incorporate non-sound generating activities between identified fine activities to predict the coarse activity. For example, predicting the pattern of hammering, nailing, sawing (i.e., fine activities) can yield to the prediction of frame building for a wall (i.e., coarse activity). For the active and idle time tracking, the primary limitation was that some activities such as painting, sanding has the higher potential of not generating enough vibration to distinguish between active and idle state. Future research will be extended for other workstations where little or no vibration is generated by utilizing computer vision and machine hearing techniques.

CHAPTER 5: OPTIMIZING LABOR ALLOCATION IN MODULAR CONSTRUCTION FACTORY

The content of Chapter 5 is an adapted version of the following manuscripts:

Rashid, K., Louis, J. & Swanson, C., (2020). "Optimizing labor allocation in modular construction factory using discrete-event simulation and genetic algorithm", Proceeding of the Winter Simulation Conference, Orlando, FL. DOI: 09/WSC48552.2020.9383867

5.1. Introduction

This chapter presents a resource allocation framework combining a discrete event simulation (DES) model and a genetic algorithm (GA) to facilitate data-driven decision-making. The DES model simulates the process of constructing modular units in the factory, and the GA optimizes the number of the worker at different workstations yielding to minimize makespan. A case study with a real-world modular construction factory showed that optimizing the assignment of available workers can reduce the makespan by up to 15%.

While it is not explicitly implemented currently, it is assumed that the results of Chapter 4 will be used to periodically update the DES model-based GA optimization to enable a near real-time decision-making framework for modular construction. This study demonstrates the potential of the proposed method as a practical tool to optimize resource allocation in uncertain work environments in modular construction factories.

5.2. Background

The construction process in modular factories closely resembles a manufacturing production line, where different workstations are dedicated to a specific type of activity (e.g., building floors and walls, installing walls, installing insulation, etc.), and through which the modular unit would travel travels through. Thus each component of the modular unit (e.g., wall, floor, ceiling, etc.), as well as the modular unit spends a different amount of time (i.e., cycle time) at each station based on their particular design specifications. An important factor that has a direct impact on the cycle time is the number of workers working at the station, which is referred to as ‘labor allocation’ in this chapter. Thus, effective and data-driven labor allocation at various stations is a significant factor in improving productivity and maximizing the benefits of modular construction.

To that end, this chapter proposes an optimization framework for labor allocation in modular construction factories by utilizing discrete event simulation (DES) and genetic algorithm (GA). The DES methodology is used to model and simulate the process workflow while the GA searches for the labor allocation to workstations that results in the optimal solution (i.e., maximum production rate or minimum makespan) in the process workflow. The following section discusses previous studies related to simulation modeling and GA-based optimization in construction.

5.2.1. Simulation modeling for modular construction

In the context of this research, simulation modeling is the process of creating and analyzing a virtual model of a real-world process to predict and forecast its performance. Simulation modeling has been widely explored for on-site and off-site construction processes in several previous studies (Afifi et al. 2017; Akhavian and Behzadan 2013; AlDurgham and Barghash 2008; Altaf et al. 2015b, 2018; Hammad et al. 2002; Jeong et al. 2011; Louis et al. 2014; Louis and Dunston 2016a; Zhang

2004). Altaf et al. (2018) proposed an integrated production planning and control system for panelized home building using DES and radio frequency identification (RFID)- based tracking. A discrete and continuous simulation approach was also explored to optimize the production of modular construction elements (Afifi et al. 2017). AlDurgham and Barghash (2008) proposed a simulation-based approach to facilitate decision-making for planning layout, material handling, scheduling, and manufacturing processes and resources for off-site house building. Lu and Olofsson (2014) proposed a framework consisting of building information modeling (BIM) and DES to enable the integration of DES in the planning and follow-up of construction activities.

5.2.2. Genetic algorithm for optimization

Genetic algorithms (GAs) are optimization techniques that are based on the principles of Darwinian evolution which simulate biological evolution through stochastic search techniques (Holland 1975). GAs has been used for optimizing simulation models in construction, as well as in other fields. Yang et al. (2016) developed a flow shop scheduling optimization model for multiple production lines for precast production. In another effort, an adaptive GA was presented for resource-leveling as a flexible decision support system to enable practitioners to choose a feasible solution (Ponz-Tienda et al. 2013). GA, analytic hierarchy process, and computer simulation were integrated for optimization of operator allocation in the cellular manufacturing process (Azadeh et al. 2014). A combination of GA with simulated annealing (SA) was also adopted for generic multi-project scheduling optimization with multiple resources constrain in complex construction projects (Chen and Shahandashti 2009).

Leu et al. (2000) presented a prototype of a decision support system for construction resource-leveling using GA to achieve an optimal or near-optimal combination of multiple construction resources.

5.3. Research gaps and point of departure

Based on the review of the literature conducted, the following gaps are identified and targeted in this chapter:

(1) There is a lack of literature related to the optimization of resources in modular construction factories.

(2) It is not known how DES modeling and GA can be coupled together to provide decision-making support in modular construction factories.

Considering the above gaps in knowledge, the goal of this research is to develop a framework to automatically optimize workers at various workstations in modular construction factories by coupling DES modeling and GA. This research goal is accomplished in this chapter through the pursuit of these specific research objectives:

(1) Develop simulation model of high-level activities in modular construction factory.

(2) Optimize the labor allocation at different workstations using a genetic algorithm (GA) and the simulation model.

(3) Compare productivity of the operation between current labor allocation vs. optimized labor allocation.

5.4. Methodology

The objective of this study is to obtain the optimal number of workers at each workstation of the modular factory to minimize the makespan, which is defined as the amount of time each unit spends in the production line from start to end. Thus the two primary components of the proposed methodology are a DES model to simulate the process of the factory, and a GA to optimize the number of workers that yield the minimum makespan of the modular units. The proposed methodology is illustrated in Figure 5.1.

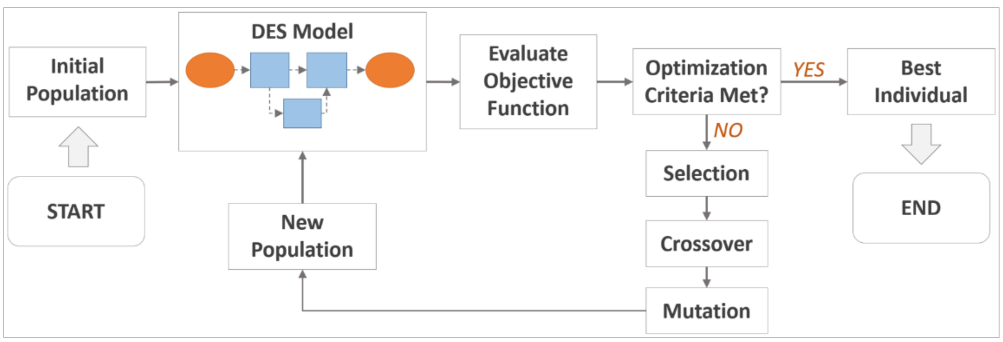


Figure 5.1. Overview of the proposed methodology

As discussed previously, the methodology consists of two primary components: DES and GA. The DES model is created by modeling the interdependencies between the workstations. The duration of the activities at the workstations is modeled as a factor of the number of workers working at the corresponding stations using a linear relationship. The output of the simulation is the average makespan of the modular unit in 100,000 minutes of simulation time. The number of workers at the workstations is the input variable, and minimizing the makespan is the objective function of the GA. As shown in Figure 1, first an initial population (i.e., a vector containing number of workers in each workstation) is selected randomly and passed to the DES model.

The DES model simulates the process of constructing modular units in the factory and average makespans for units are calculated. Then the objective function (i.e., to minimize makespan) is evaluated for every one of the population to check whether it meets the optimization criteria (i.e., a threshold value for minimum makespan). If yes, this best individual from the population is selected as the best solution. If no, two pairs of individuals (i.e., vectors containing the number of workers) are selected as parents based on their fitness score (i.e., minimum makespan). For each pair of the parents to be mated, a crossover point is chosen at random from within their genes (i.e., index of the worker vector), and the offspring exchanges the genes of parents among themselves until the crossover point. After new offspring (i.e., a new vector of the number of workers) is created, some of their genes can be subjected to a mutation where some of the genes in the offspring can be flipped. An illustration of crossover and mutation is shown in Figure 5.2.

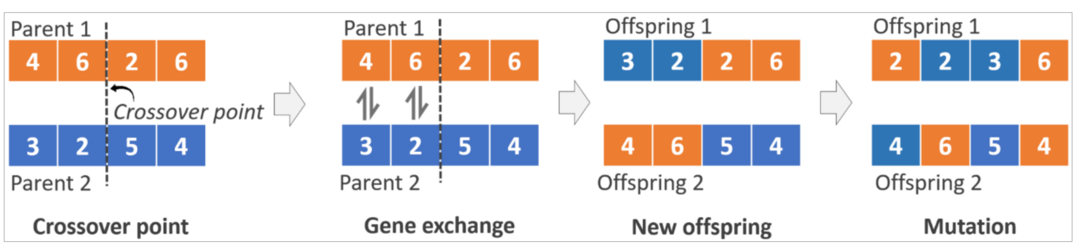


Figure 5.2. Crossover and mutation of GA

After a new population is generated, they are again passed to the DES model to calculate makespan for construction modular units. This process of selection, crossover, and mutation continues until the optimization criteria are met. Pseudocode for the GA can be expressed as:

```
START
Generate the initial population
Compute fitness
REPEAT
    Selection
    Crossover
    Mutation
    Compute fitness
UNTIL population has converged
STOP
```

This methodology was validated using the data from a real-world modular construction factory, and the case study is presented in the following section.

5.5. Case Study and Results

To validate the proposed methodology, a real-world modular construction factory was selected. This factory makes volumetric modular units for projects like multi-family housing that are then shipped and set up on-site. There are several workstations in the factory dedicated to specific activities. The workstations in that factory can be divided into two categories: off-line stations and online stations. The panelized components of the unit, such as walls, floors, and ceilings are built from raw material in the off-line stations. The online stations are part of the assembly line for the volumetric unit, where various pre-made components (some from the off-line stations) are added and assembled to the modular unit.

A schematic floor plan with the major workstations of the factory is shown in Figure 3. Three main off-line workstations clusters are *Partial Wall*, *Long Wall*, and *Ceiling* stations. Some of those off-line stations are further divided into several smaller stations dedicated to separate activities. For example, in the ceiling workstation, first, the ceiling frame and drywall are put and moved for rough plumbing and electrical station. Eventually, the final ceiling is built and moved to the online station to assemble to the modular unit. Each of the workstations (both online and off-line) are denoted with the number of worker and cycle time in Figure 5.3. For example, typically 5 workers are placed to the first online station, *Floor Build*, and it takes about 260 minutes to build a 50 feet floor. Cycle time and the typical number of workers were acquired from actual time study, expert opinion, as well as from the experience of floor manager of the factory.

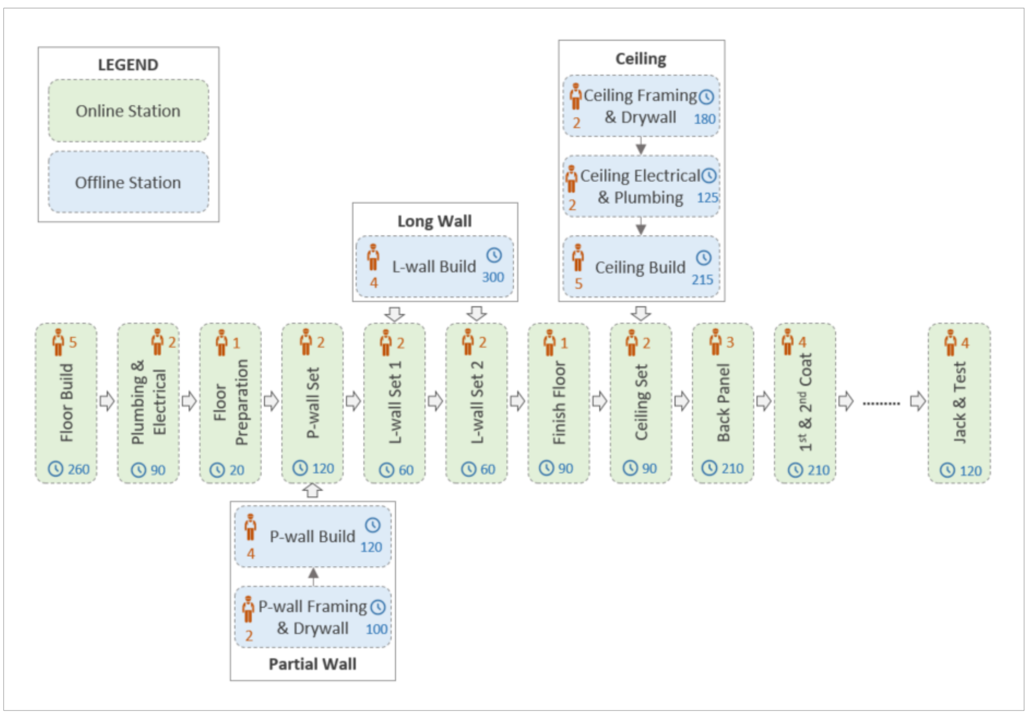


Figure 5.3. A schematic of the floor plan of the modular factory

After gathering the cycle time and worker data, a DES model of the factory floor was developed using the *SimEvent* tool of *MathWorks*. Figure 5.4 shows the diagram of the DES model, where the three major off-line workstations (i.e., *Partial Wall*, *Long Wall*, and *Ceiling*) are denoted by rectangular borders. It should be noted that there are two different *Long Wall* (denoted as *Long Wall 1* and *Long Wall 2*) stations in the factory to build long walls of two sides of the modular units. The number “1” denoted at each activity represents the queue capacity of each station. In the factory, each station can only handle one component at a time. There are some surge spaces on the factory floor which work as queues with a FIFO (i.e., first-in-first-out) queue policy. Several assumptions were made while developing the DES model, such as the transfer of material from one station to another is instantaneous, there are no constraints for the workers to move from one station to another, every worker is eligible to work in any station, etc. The assumptions were made because the focus of this study is on the aspect of labor allocation.

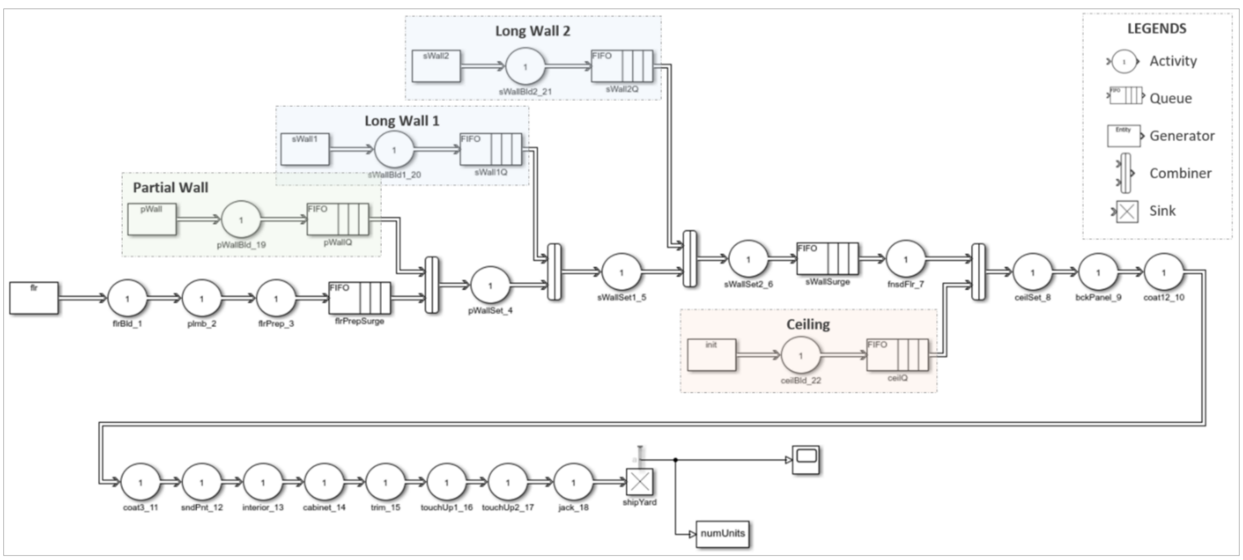


Figure 5.4. The DES model of the modular factory

The cycle times of the activities were converted to worker minutes and were set up as a function of the number of workers. A linear relationship was assumed between the number of workers at a station and its cycle time based on input from the factory manager. A genetic algorithm was developed where the variables were the number of workers at each workstation, and the objective was to minimize the makespan. The maximum number of workers in the factory was 74, which was the same as currently placed in the factory. The lower and upper bound of the worker number was set as one and eight, respectively. At each iteration of the simulation, the GA module sent an array to the DES model representing the number of workers at each workstation, the simulation model calculated the makespan and sent it to the GA module. Progressively the GA minimized the makespan and the simulation progress is shown in Figure 5. The GA was run for 1000 generations, each generation containing 30 populations. Figure 5.5 shows that the penalty value (i.e., average makespan) plateaued after 460 generations.

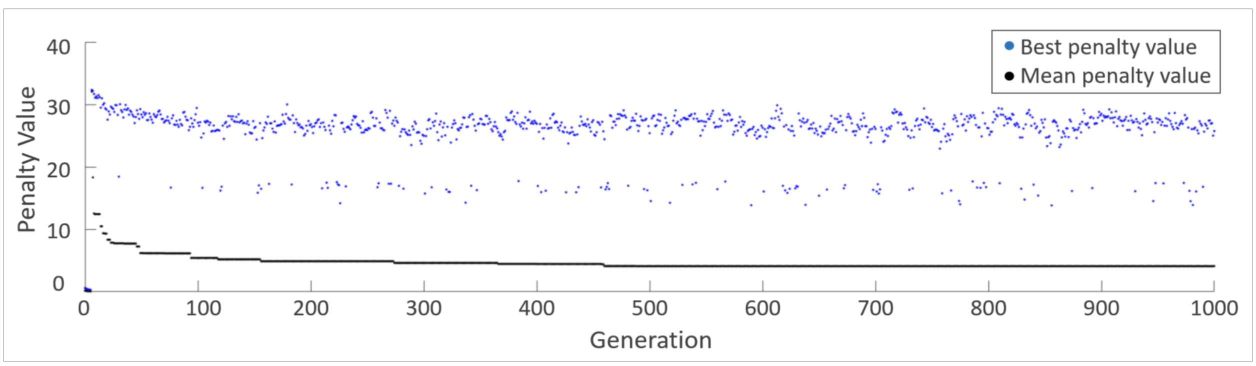


Figure 5.5. Optimization of the makespan using genetic algorithm

The duration of each station was assumed to be a triangular distribution with 15% upper and lower bound. The average makespan was calculated for three different combinations of the number of workers. The simulation model was run 100 times using the number of workers in the factory (i.e., as is), an optimized combination using 74 workers (i.e., GA Opt_74), as well as an optimized combination using 100 workers (GA Opt_100). Figure 5.6 shows the boxplot containing the makespan distribution of each of the three combinations. The first boxplot shows the makespan with current "as is" labor allocation in the factory. The second boxplot is the makespan with an optimized number of workers by GA with a maximum 74 number of workers in the factory. The last boxplot illustrates the makespan with a maximum of 100 workers in the factory.

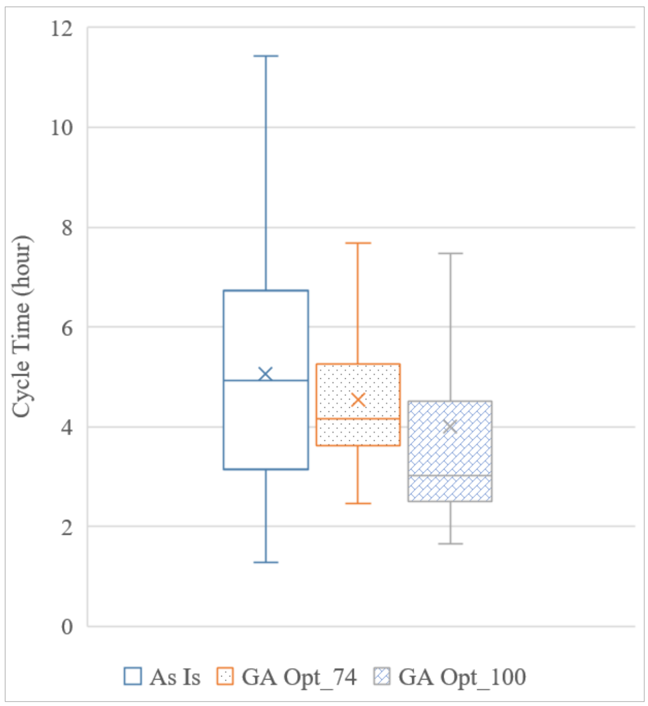


Figure 5.6. The DES model of the modular factory

The median makespan for "as is" is 4.92 hours, "GA optimized with 74 workers" is 4.16 hours, and "GA optimized with 100 workers" is 3.02 hours. Thus, this analysis shows that the makespan can be reduced by about 15% with the same total number of workers currently situated in the factory, just by shuffling their numbers in a couple of stations. Moreover, if a decision is made to increase the total number of workers from 74 to 100, the makespan can be reduced by about 38%. Table 5.1 shows the result of the optimized number of workers at different workstations.

Table 5.1. Optimized number of workers at each workstation

Station Name	As-Is	GA Opt_74
Floor Build	7	8
Plumbing	3	3
Floor Prep	2	2
P-Wall Set	2	2
S-Wall Set 1	2	2
S-Wall Set 2	2	2
Finished Floor	2	2
Ceiling Set	2	2
Back Panel	2	4
1st and 2nd Coat	5	6
3rd Coat	3	4
Sand and Paint	6	6
Interior	2	2
Cabinet	6	8
Trimming	2	2
Touch Up 1	4	3
Touch Up 2	6	3
Jack	4	2
P-Wall Build	4	3
S-Wall Build 1	2	2
S-Wall Build 2	2	2
Ceiling Build	4	4

Stations requiring a change in the number of workers are highlighted in the table. The analysis is suggesting adding workers to *Floor Build* (1 worker), *Back Panel* (2 workers), *1st and 2nd Coat* (1 worker), *3rd Coat* (1 worker), and *Cabinet* (2 workers) stations, and reduce workers from *Touch Up1* (1 worker), *Touch Up 2* (3 workers), and *Jack* (2 workers) station. However, in both cases, the total number of workers in the factory remains the same (74 workers).

5.6. Conclusion and future work

This study presents a methodology for optimizing the number of workers at different workstations in a modular construction factory. This consists of a DES model and a GA to optimize the DES model. A vector consisting of the number of workers is sent to the DES model to calculate the average makespan, and the GA simultaneously tried to minimize the average makespan by performing selection, crossover, and mutation until the performance of the algorithm plateaued. A real-world modular construction factory was selected to validate the proposed methodology. In particular, there were 22 different workstations in the factory dedicated to various activities. The entire assembly line process of the modular construction factory was modeled using DES, and the GA optimization showed a makespan reduction of 15%. Specifically, if the total number of workers in the factory remains the same (i.e., 74), reallocating the number of workers based on the results of the analysis can yield 15% makespan savings. If the number of workers in the entire factory is increased to 100, the optimization showed a 38% reduction in makespan.

The real-world case study illustrated that the proposed approach could help management to optimize worker allocation in the complex modular construction factory. The dynamic approach of labor allocation presented in this chapter can eliminate the limitations of traditional CPM-based resource allocation by updating the DES model at the desired interval. The integration of worker tracking technologies using an indoor positioning system (IPS) or computer vision with the DES model in the proposed system can unleash the true potential of a dynamic data-driven decision support system. The primary limitations of the DES model stem from the assumptions made in terms of instantaneous transfer of components, linear relationship between duration of an activity and the number of workers, and the same expertise level for all the workers. In the future, the abovementioned dependencies will be added to the DES model for a more realistic simulation of the process. Moreover, future research will be directed to develop multi-objective optimization, where other criteria (e.g., wait time, worker expertise, etc.) can be included in the objective function. An adaptive GA can be explored for real-time dynamic optimization of worker allocation in the future.

CHAPTER 6: CONCLUSION, LIMITATIONS, AND FUTURE RESEARCH

The overarching goal of this dissertation was to **advance the body of knowledge and make practical contributions to integrate emerging technologies with project monitoring for data-driven decision-making.** Two major objectives are identified to achieve this goal, and they are

1. Automatically identify states (i.e., location and activity) of the resources in the construction site.

2. Develop dynamic simulation tools to analyze and optimize the operations. To attain the research goal and objectives, the following questions were developed to guide the overall development and outcome of this study:

Q1. What type of consumer-grade sensors can be used on construction sites to ensure a safe and reliable source of data collection?

Q2. What type of data analytics techniques is appropriate to automatically identify activities from the resources?

Q3. What are the key considerations that ensure optimum performance of the activity identification framework?

Q4. Can simulation tools be used to provide dynamic predictions using the outcomes of the activity identification framework?

Q5. What type of feedback can be provided from the monitoring and simulation for timely and effective decision-making?

To answer the research questions, and achieve the research goal and objectives, four research activities were conducted, and are each presented in single chapters from Chapters 2 to 5. Chapters 2 and 3 addressed Research Objectives 1 and 2 for heavy civil operations, while Chapters 4 and 5 addressed Research Objectives 1 and 2 for modular construction operations. The methods developed and implemented, main conclusions, research limitations, and future directions are summarized in this section. Finally, the overall conclusions and contributions of the entire research study are discussed for future research.

6.1. Automated Activity Identification for Heavy Civil Operations (Chapter 2)

The research presented in Chapter 2 was performed to accomplish the objective of enabling automated activity identification for heavy civil operations by answering the following research questions related to (1) exploring consumer-grade sensors for automated and reliable data collection from construction operations, (2) appropriate analytical tools for automated activity identification in construction, and (3) various technical and physical considerations to optimize the performance of automated activity identification of construction operations. These three questions were answered in this chapter focusing on heavy civil operations.

To answer these questions, this chapter developed a deep learning-based activity identification framework to automatically identify activities of heavy civil equipment from earthmoving sites using inertial measurement units (IMUs). Several IMUs were attached to different articulated components of the equipment (e.g., bucket, boom, stick of the excavator). Vibration and orientation data were collected, processed, and segmented using sliding window techniques. These segmented data were used as inputs to a long short-term memory (LSTM) network to train the model. Moreover, several data augmentation techniques were implemented to generate synthetic training data to reduce manual field data collection efforts. Finally, test data were used to evaluate the performance of the trained model to identify various activities. In doing so, various sensitivity analyses were performed to explore physical and technical parameters that optimize the performance of the trained model. The developed methodology was validated for three major types of heavy civil equipment: excavator, front-end loader, and dump truck. The major conclusions from this research are:

(1) As opposed to the traditional machine learning classification algorithms, the LSTM network contains long-term temporal dependency of the training data between consecutive time steps.

(2) The LSTM network eliminated the necessity of manual feature extraction, which is limited to human domain knowledge. Instead, the deep network automatically learned high-level representative features from the raw training data.

(3) Implementation of data augmentation eliminated the necessity of collecting a large volume of training data from the construction site. This improves the practicality of such classification techniques for temporary and transient construction operations.

(4) Synthetic training data removed bias in the trained model due to the imbalanced volume of training data. For example, if there is little training

data of one specific class compared to other classes, the trained model might be biased towards that class.

(5) The location with the highest degree of movement is the best place to attach the sensor for activity identification purposes.

Research presented in this study shows the potential of consumer-grade sensors and deep learning algorithms to automatically identify activities from heavy civil equipment. However, there are several limitations in this study, which include:

(1) Training a deep learning algorithm requires lots of training data, which sometimes is challenging to collect from an active construction site. Even though data augmentation techniques showed promising results in generating synthetic training datasets, further study is required to investigate the acceptable amount of field data collection without compromising the performance of the trained model.

(2) Field data were collected from three different excavator models, one front-end loader, and two types of dump-truck models. In practice, there are lots of variety in the sizes and shapes of this equipment. So, further study is required to develop a generalized model incorporating different sizes and shapes of the equipment into the model for practical use.

6.2. Automated Decision-making for Heavy Civil Operations (Chapter 3)

The research presented in Chapter 3 was performed to accomplish the objective of developing a dynamic simulation tool for analyzing and optimizing heavy civil operations. This objective was accomplished by answering the research questions related to (1) exploring the capacity of simulation tools to perform dynamic productivity estimation, and (2) appropriate feedback provided for effective decision-making. These two questions were answered in this chapter focusing on heavy civil operations.

To answer the questions mentioned above, this chapter provides a framework that integrates automatically collected field data from IMUs mounted on heavy equipment and processes them for use as updated input for simulation models to estimate dynamic productivity which presents the operation realistically. Activities (e.g., loading, hauling, etc.) are identified from the collected data and the cycle times are calculated using machine learning methods. The cycle time distributions are updated using Bayesian updating and used as inputs in the simulation model for more accurate productivity prediction. A real-world earthmoving site is used in the case study to validate the framework. This framework will help the decision-making process by realistically reflecting the field condition using the most up-to-date information in the simulation model. The major findings of this study were:

(1) Simulation modeling and Bayesian updating methods can be used to update the productivity estimation of earthmoving projects during the construction phase.

(2) The proposed framework can provide aid to the managers in deciding on adding or removing equipment from the fleet to optimize productivity.

The real-world case study in this research illustrated that the proposed approach could help management to optimize fleet composition in heavy civil projects. However, there are certain limitations in this study, and they are:

(1) Several assumptions were made during developing the DES model, such as only interaction between the excavator and dump trucks were modeled. In reality, other equipment such as the loader, grader, etc. are also in operation in earthmoving sites. Thus, further work is required to completely comprehend the complex interaction among all those different equipment during the construction phase.

(2) Even though this provides a platform for managers to optimize productivity by making changes to their equipment fleet, no actual analysis was conducted to demonstrate how this manual/automated optimization process can work in actuality. Future work is envisioned utilizing reinforcement learning (RL) – based policy optimization techniques to automatically optimize the fleet composition and provide the manager with shortlisted options.

6.3. Automated Activity Recognition for Modular Construction (Chapter 4)

The research presented in Chapter 4 was performed to accomplish the objective of enabling automated activity identification and idle time tracking for prefabricated construction operations by answering the following research questions related to (1) exploring consumer-grade sensors for automated and reliable data collection from construction operations, (2) appropriate analytical tools for automated activity identification and idle time tracking in construction, and (3) various technical and physical considerations to optimize the performance of automated activity identification of construction operations. These three questions were answered in this chapter focusing on modular construction operations.

To answer the questions mentioned above, this study proposed and validated an automated audio-based manual activity identification framework for modular construction. Also, the IMU-based framework successfully tracks idle and active time in different workstations. Audio and IMU data were collected from an actual modular construction factory. Both audio and IMU data were processed, segmented, and used as inputs to machine learning and deep learning models. Finally, test data were used to evaluate the performance of the trained model. In developing the framework particular focus was given towards engineering appropriate features, which essentially is a paramount step in ensuring robust model training and validation.

The major findings from this study are:

(1) Audio can be used as a reliable data source in identifying manual activities inside a modular construction factory.

(2) Audio augmentation techniques significantly improve the performance of the activity identification model.

(3) A systematic approach to engineer appropriate features from all the four different domains (e.g., time-, frequency-, cepstral-, and wavelet) is required to maximize the performance of the prediction model.

(4) Active and idle times can be automatically and accurately tracked using the vibration generated from work performed in the workstations.

While the proposed model successfully can identify manual activities and active, idle time in a modular construction factory, there are several limitations to the work presented. The limitations are:

(1) The scope of this research was to identify single activities happening at any point in time. However, in real-world conditions, multiple activities happen simultaneously, thereby generating a layered audio file. This problem falls under the multi-label time series classification domain. Further work is required to investigate the use of microphone arrays and beamforming techniques to identify sources of sound and simultaneous identification of multiple activities.

(2) To derive meaningful information from the factory, it is essential to identify complex activities, such as building walls, assembling ceilings, etc. While this study provides the first pillar on audio-based activity identification, future work is required to combine sequential manual actions (e.g., hammering, nailing, etc.) to predict complex activities.

(3) The consumer-grade IMU used in this study to collect vibration data from the workstations was not sensitive towards low-amplitude vibrations. These sensors might miss activities like mudding, painting, finishing, those do not generate high-amplitude vibrations. Thus, further investigation is required to explore the usability of high-fidelity sensors to capture smaller vibrations and track active times from them.

6.4. Automated Decision-making for Modular Construction (Chapter 5)

The research presented in Chapter 5 was performed to accomplish the objective of developing a dynamic simulation tool for analyzing and optimizing operation in modular construction operations. This objective was accomplished by answering the research questions related to (1) exploring the capacity of simulation tools to perform dynamic productivity estimation, and (2) appropriate feedback provided for effective decision-making. These two questions were answered in this chapter focusing on modular construction operations.

To answer these questions, this study utilizes the capability of discrete-event simulation (DES) to model the high-level operations of a modular construction factory. Cycle times were used from the outputs of chapter 4, as well as from the manual cycle time study. Dependencies of productivity on available labor were also modeled and a genetic algorithm (GA) was developed to optimize the simulation input parameter (i.e., labor allocation at workstations). The major findings of this study were:

(3) Simulation modeling and optimization algorithms can be successfully used to optimize labor allocation in the modular construction factory.

(4) The proposed framework can provide aid to the managers in deciding on adding or removing workforces from different workstations.

The real-world case study in this research illustrated that the proposed approach could help management to optimize worker allocation in the complex modular construction factory. However, there are certain limitations in this study, and they are:

(1) Several assumptions were made during the development of the DES model. Such as instantaneous transfer of components (i.e., no time lag was considered while moving modular components from one station to another), linear relationship between duration of an activity and the number of workers, and same expertise level for all the workers. A more

robust model can be built in the future addressing these assumptions for a more accurate prediction.

(2) The optimization process using a genetic algorithm (GA) is not computationally efficient for real-time usage. Thus, it will be challenging to implement the proposed method for dynamic prediction in the decision-making process. However, reinforcement learning (RL) – based optimization techniques can be adopted using the same principles for time-efficient thus the real-time implementation of the proposed framework.

(3) This research used a single objective optimization approach, minimizing the average makespan of the modular units. Future study is required to investigate multi-objective optimization, where other criteria (e.g., wait time, worker expertise, etc.) can be included in the objective function.

6.5. Overall research conclusion and contributions

The present research demonstrates the feasibility of consumer-grade sensors, machine learning algorithms, and simulation modeling techniques to automatically monitor construction operations, optimize them, and provide aid for data-driven decision-making. This study contributes to the body of knowledge by providing a means for automated monitoring of construction operations using emerging technologies and assessing the use of simulation modeling for data-driven decision-making. Future researchers could use the findings and insights of this study as a starting point to advance the knowledge for a connected and smart construction environment.

The study also makes a practical contribution by developing machine learning and deep learning models to automatically identify construction activities. One model uses inertial measurement units (IMUs) to track heavy civil operations and another model uses audio signals to monitor manual activities inside a modular construction factory. Moreover, a framework was also presented to optimize the operations using a discrete event simulation (DES) model, dynamic prediction, and optimization algorithms. Researchers could use the information presented in this work to develop more advanced tools and models for a similar type of construction operation. Moreover, construction practitioners could utilize the proposed models to make timely data-driven decisions.

6.6. Recommendations for future research

The physical and technical parameters for automated activity identification of construction operations presented in this study are expected to set the foundation for subsequent and future work on digitization, and a data-driven approach in construction operations to improve efficiency. Further studies are required to address the limitations of the current study, as well as push the boundaries of the application of emerging technologies in construction.

As for the activity identification of heavy civil equipment, limited case studies were conducted. Future studies are anticipated to apply and validate the proposed framework for other types of construction equipment. Moreover, wireless technologies, the internet of things (IoT), and database management systems could be integrated to implement the proposed system in real-time. An accurate and reliable activity identification model could be used for several other practical purposes, such as automated productivity monitoring, safety, environmental assessment, and applications in AR/VR visualization.

As for the developed audio-based activity identification model for modular construction, microphone arrays and audio source localization techniques could be used to predict multiple activities simultaneously. More high-end and expensive sensors could be used to investigate the validity of the proposed model for activities like mudding, painting, finishing, etc. that generate low-magnitude vibrations.

Lastly, future studies could be performed to tie all the different components described in this study by using wireless technology, IoT, and database management system for real-time data-driven decision making.

REFERENCES

Afifi, M., Al-Hussein, M., Abourizk, S., Fotouh, A., and Bouferguene, A. (2017). "Discrete and Continuous Simulation Approach to Optimize the Productivity of Modular Construction Element." *Proceedings of the 33rd International Symposium on Automation and Robotics in Construction (ISARC)*, (July).

Ahn, C. R., Lee, S., and Peña-Mora, F. (2015). "Application of Low-Cost Accelerometers for Measuring the Operational Efficiency of a Construction Equipment Fleet." *Journal of Computing in Civil Engineering*, 29(2), 04014042.

Akhavian, R., and Behzadan, A. H. (2012a). "An integrated data collection and analysis framework for remote monitoring and planning of construction operations." *Advanced Engineering Informatics*, 26(4), 749–761.

Akhavian, R., and Behzadan, A. H. (2012b). "Remote monitoring of dynamic construction processes using automated equipment tracking." *Construction Research Congress 2012: Construction Challenges in a Flat World, Proceedings of the 2012 Construction Research Congress*, H. Cai, A. Kandil, M. Hastak, and P. Dunston, eds., American Society of Civil Engineers, West Lafayette, Indiana, 1360–1369.

Akhavian, R., and Behzadan, A. H. (2013). "Knowledge-Based Simulation Modeling of Construction Fleet Operations Using Multimodal-Process Data Mining." *Journal of Construction Engineering and Management*, 139(11), 04013021.

Akhavian, R., and Behzadan, A. H. (2015). "Construction equipment activity recognition for simulation input modeling using mobile sensors and machine learning classifiers." *Advanced Engineering Informatics*, 29(4), 867–877.

Akhavian, R., and Behzadan, A. H. (2016). "Smartphone-based construction workers' activity recognition and classification." *Automation in Construction*, 71(Part 2), 198–209.

Akhavian, R., Brito, L., and Behzadan, A. (2015). "Integrated Mobile Sensor-Based Activity Recognition of Construction Equipment and Human Crews." *Conference on Autonomous and Robotic Construction of Infrastructure*, Ames, Iowa, 1–20.

Alazzaz, F., and Whyte, A. (2014). "Uptake of Off-site Construction : Benefit and Future Application." *International Journal of Civil, Environmental, Structural, Construction and Architectural Engineering*, 8(12), 1168–1172.

AlDurgham, M. M., and Barghash, M. A. (2008). "A generalised framework for simulation-based decision support for manufacturing." *Production Planning and Control*, 19(5), 518–534.

Alías, F., Socoró, J. C., and Sevillano, X. (2016). "A review of physical and perceptual feature extraction techniques for speech, music and environmental sounds." *Applied Sciences*, 6(5).

Altaf, M. S., Bouferguene, A., Liu, H., Al-Hussein, M., and Yu, H. (2018). "Integrated production planning and control system for a panelized home prefabrication facility using simulation and RFID." *Automation in Construction*, 85(September 2017), 369–383.

Altaf, M. S., Liu, H., Al-Hussein, M., and Yu, H. (2015a). "Online simulation modeling of prefabricated wall panel production using RFID system." *Winter Simulation Conference*, 1576–1580.

Altaf, M. S., Liu, H., Zhang, Y., Al-Hussein, M., and Bouferguene, A. (2015b). "Discrete-event simulation modelling of prefabricated wall production line." *27th European Modeling and Simulation Symposium, EMSS 2015*, (c), 234–239.

Ang, A. H.-S., and Tang, W. H. (1975). *Probability Concepts in Engineering Planning and Design, Basic Principles*. Wiley, New York.

Azadeh, A., Asadzadeh, S. M., Mehrangohar, M., and Fathi, E. (2014). "Integration of genetic algorithm, analytic hierarchy process and computer simulation for optimisation of operator allocation in manufacturing systems with weighted variables." *International Journal of Logistics Systems and Management*, 17(3), 318–339.

Azimi, R., Lee, S., Abourizk, S. M., and Alvanchi, A. (2011). "A framework for an automated and integrated project monitoring and control system for steel fabrication projects." *Automation in Construction*, 20(1), 88–97.

Bao, R., Sadeghi, M. A., and Golparvar-Fard, M. (2016). "Characterizing Construction Equipment Activities in Long Video Sequences of Earthmoving Operations via Kinematic Features." *Construction Research Congress*, ASCE, San Juan, Puerto Rico, 849–858.

Barbosa, F., Woetzel, J., Mischke, J., Ribeirinho, M. J., Sridhar, M., Bertram, N., and Brown, S. (2017). *Reinventing Construction: A Route To Higher Productivity*. *McKinsey & Company*.

Behzadan, A. H., and Kamat, V. R. (2012). “A Framework for Utilizing Context-Aware Augmented Reality Visualization in Engineering Education.” *International Conference on Construction Application of Virtual Reality*, 292–299.

Bengio, Y. (2013). “Deep learning of representations: Looking forward.” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 7978 LNAI, 1–37.

Bogert, B. P. (1963). “The quefrency analysis of time-series for echoes; Cepstrum, pseudio-autocovariance, cross-cepstrum and saphe craking.” *Time series analysis*, 209–243.

Carbonari, A., Giretti, A., and Naticchia, B. (2011). “A proactive system for real-time safety management in construction sites.” *Automation in Construction*, 20(6), 686–698.

Chan, W.-H., and Lu, M. (2008). “Materials Handling System Simulation in Precast Viaduct Construction: Modeling, Analysis, and Implementation.” *Journal of Construction Engineering and Management*, 134(4), 300–310.

Charalambous, C. C., and Bharath, A. A. (2016). “A data augmentation methodology for training machine/deep learning gait recognition algorithms.” 1–12.

Chen, J., Qiu, J., and Ahn, C. (2017). “Construction worker’s awkward posture recognition through supervised motion tensor decomposition.” *Automation in Construction*.

Chen, P. H., and Shahandashti, S. M. (2009). “Hybrid of genetic algorithm and simulated annealing for multiple project scheduling with multiple resource constraints." *Automation in Construction*, Elsevier B.V., 18(4), 434–443.

Chen, S. M., Griffis, F. H., Chen, P. H., and Chang, L. M. (2012). "Simulation and analytical techniques for construction resource planning and scheduling." *Automation in Construction,* 21(1), 99–113.

Cheng, B., Wei, Y., Zhang, W., Zhou, X., Chen, H., Huang, L., Huang, J., and Kang, X. (2020). "Evolutionary game simulation on government incentive strategies of prefabricated construction: A system dynamics approach." *Complexity*, 2020.

Cheng, C.-F., Rashidi, A., Davenport, M. A., and Anderson, D. V. (2017). "Activity analysis of construction equipment using audio signals and support vector machines." *Automation in Construction*, 81, 240–253.

Cheng, C.-F., Rashidi, A., Davenport, M. A., and Anderson, D. V. (2019). "Evaluation of Software and Hardware Settings for Audio-Based Analysis of Construction Operations." *International Journal of Civil Engineering*, Springer International Publishing, 17(9), 1469–1480.

Cheng, T., and Teizer, J. (2013). "Real-time resource location data collection and visualization technology for construction safety and activity monitoring applications." *Automation in Construction*, 34, 3–15.

Cho, C., Lee, Y.-C., and Zhang, T. (2017). "Sound Recognition Techniques for Multi-Layered Construction Activities and Events." *Computing in Civil Engineering 2015*, 667–674.

Chu, S., Narayanan, S., and Kuo, C.-C. J. (2009). "Environmental Sound Recognition

With Time-Frequency Audio Features." *IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING*, 17(6).

Chung, T. H., Mohamed, Y., and AbouRizk, S. (2006). "Bayesian Updating Application into Simulation in the North Edmonton Sanitary Trunk Tunnel Project." *Journal of Construction Engineering and Management*, 132(8), 882–894.

D'Innocente, A., Carlucci, F. M., Colosi, M., and Caputo, B. (2017). "Bridging between computer and robot vision through data augmentation: A case study on object recognition." *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 10528 LNCS(figure 2), 384–393.

Demiralp, G., Guven, G., and Ergen, E. (2012). "Analyzing the benefits of RFID technology for cost sharing in construction supply chains: A case study on prefabricated precast components." *Automation in Construction*, Elsevier B.V., 24, 120–129.

Ding, J., Chen, B., Liu, H., and Huang, M. (2016). "Convolutional Neural Network with Data Augmentation for SAR Target Recognition." *IEEE Geoscience and Remote Sensing Letters*, 13(3), 364–368.

Ding, L., Fang, W., Luo, H., Love, P. E. D., Zhong, B., and Ouyang, X. (2018). "A deep hybrid learning model to detect unsafe behavior: Integrating convolution neural networks and long short-term memory." *Automation in Construction*.

Dong, S., and Kamat, V. R. (2013). "SMART: scalable and modular augmented reality template for rapid development of engineering visualization applications." *Visualization in Engineering*, 1(1), 1–17.

Du, J., Jing, H., Castro-Lacouture, D., and Sugumaran, V. (2019). "Multi-agent simulation for managing design changes in prefabricated construction projects." *Engineering, Construction and Architectural Management*, 27, 270–295.

El-Omari, S., and Moselhi, O. (2011). "Integrating automated data acquisition technologies for progress reporting of construction projects." *Automation in Construction*, 20(6), 699–705.

Ergen, E., Akinci, B., East, B., and Kirby, J. (2007). "Tracking Components and Maintenance History within a Facility Utilizing Radio Frequency Identification Technology." *Journal of Computing in Civil Engineering*, 21(1), 11–20.

Forestier, G., Petitjean, F., Dau, H. A., Webb, G. I., and Keogh, E. (2017). "Generating synthetic time series to augment sparse datasets." *Proceedings - IEEE International Conference on Data Mining, ICDM*, 2017-Novem, 865–870.

Fu, Z., Lu, G., Ting, K. M., and Zhang, D. (2011). "A survey of audio-based music classification and annotation." *IEEE Transactions on Multimedia*, IEEE, 13(2), 303–319.

García-Hernández, A., Galván-Tejada, C. E., Galván-Tejada, J. I., Celaya-Padilla, J. M., Gamboa-Rosales, H., Velasco-Elizondo, P., and Cárdenas-Vargas, R. (2017). "A similarity analysis of audio signal to develop a human activity recognition using similarity networks." *Sensors*, 17(11), 1–15.

Gather, U., and Fried, R. (2002). *Robust Preprocessing of Time Series with Trends.*

Informatik bewegt.

Gibb, A., and Isack, F. (2003). "Re-engineering through pre-assembly: client expectations and drivers." *Building Research and Information*, 31(2), 146–160.

Goh, M., and Goh, Y. M. (2019). "Lean production theory-based simulation of modular construction processes." *Automation in Construction*, Elsevier, 101(December 2018), 227–244.

Golparvar-Fard, M., Heydarian, A., and Niebles, J. C. (2013). "Vision-based action recognition of earthmoving equipment using spatio-temporal features and support vector machine classifiers." *Advanced Engineering Informatics*, 27(4), 652–663.

Gong, J., and Caldas, C. H. (2009). "An Intelligent Video Computing Method for Automated Productivity Analysis of Cyclic Construction Operations." *Proceedings of the ASCE International Workshop on Computing in Civil Engineering*, 64–73.

Gong, J., Caldas, C. H., and Gordon, C. (2011). "Learning and classifying actions of construction workers and equipment using Bag-of-Video-Feature-Words and Bayesian network models." *Advanced Engineering Informatics*, 25(4), 771–782.

González, V., and Echaveguren, T. (2012). "Exploring the environmental modeling of road construction operations using discrete-event simulation." *Automation in Construction*, Elsevier B.V., 24, 100–110.

Goodrum, P. M., Haas, C. T., Caldas, C., Zhai, D., Yeiser, J., and Homm, D. (2011). "Model to Predict the Impact of a Technology on Construction Productivity."

Journal of Construction Engineering and Management, 137(9), 678–688.

Graves, A., Mohamed, A. R., and Hinton, G. (2013). “Speech recognition with deep recurrent neural networks.” *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, IEEE, (6), 6645–6649.

Greff, K., Srivastava, R. K., Koutn\'\ik, J., Steunebrink, B. R., and Schmidhuber, J. (2015). “LSTM: Search Space Odyssey.” *CoRR*, abs/1503.0(10), 2222–2232.

Le Guennec, A., Malinowski, S., and Tavenard, R. (2016a). *Data Augmentation for Time Series Classification using Convolutional Neural Networks*.

Le Guennec, A., Malinowski, S., and Tavenard, R. (2016b). “Data Augmentation for Time Series Classification using Convolutional Neural Networks.” *2nd ECML/PKDD Workshop on Advanced Analytics and Learning on Temporal Data*.

Hammad, A., Senghore, O., Hastak, M., and Syal, M. (2002). “Simulation Model for Manufactured Housing Process.” *Computing in Civil Engineering*, 286–297.

Hammerla, N. Y., Halloran, S., and Plötz, T. (2016). “Deep, convolutional, and recurrent models for human activity recognition using wearables.” *IJCAI International Joint Conference on Artificial Intelligence*, 2016-Janua, 1533–1540.

Hamza, M., Shahid, S., Bin Hainin, M. R., and Nashwan, M. S. (2019). “Construction labour productivity: review of factors identified.” *International Journal of Construction Management*, Taylor & Francis, 0(0), 1–13.

Haubrick, P., and Ye, J. (2019). “Robust Audio Sensing with Multi-Sound

Classification." *International Conference on Pervasive Computing and Communications*, IEEE, Kyoto, Japan, 1–7.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). "Deep Residual Learning for Image Recognition Kaiming." ((ed.), Oxford, U.K., Pergamon Press PLC, 1989, Section 3, pp.111-120. (ISBN 0-08-036148-X)), 1–9.

Heydarian, A., Golparvar-Fard, M., and Niebles, J. C. (2012). "Automated Visual Recognition of Construction Equipment Actions Using Spatio-Temporal Features and Multiple Binary Support Vector Machines." *Construction Research Congress*, ASCE, West Lafayette, Indiana, 889–898.

Holland, J. (1975). *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arobor, Michigan.

Inoue, M., Inoue, S., and Nishida, T. (2016). "Deep recurrent neural network for mobile human activity recognition with high throughput." *Artificial Life and Robotics*, 23(2), 173–185.

Jaitly, N., and Hinton, G. E. (2013). "Vocal Tract Length Perturbation (VTLP) improves speech recognition." *Proc. ICML Workshop on Deep Learning for Audio, Speech and Language*.

Jeong, J. G., Hastak, M., Syal, M., and Hong, T. (2011). "Internal relationship modeling and production planning optimization for the manufactured housing." *Automation in Construction*, Elsevier B.V., 20(7), 864–873.

Jeong, W. S., Chang, S., Son, J. W., and Yi, J. S. (2016). "BIM-integrated construction operation simulation for just-in-time production management."

Sustainability (Switzerland), 8(11), 1–25.

Jureidini, F., Afifi, M., Zhang, R., and Al-Hussein, M. (2016). “3D / 4D Visualization Framework for Modelling off-site Productivity of Modular Construction.” *Modular and Offsite Construction (MOC) Summit*, Alberta, Canada, 83–89.

Khosrowpour, A., Niebles, J. C., and Golparvar-Fard, M. (2014). “Vision-based workface assessment using depth images for activity analysis of interior construction operations.” *Automation in Construction*.

Kim, H., Ahn, C. R., Engelhaupt, D., and Lee, S. (2018a). “Application of dynamic time warping to the recognition of mixed equipment activities in cycle time measurement.” *Automation in Construction*, 87, 225–234.

Kim, J., Chi, S., and Seo, J. (2018b). “Interaction analysis for vision-based activity identification of earthmoving excavators and dump trucks.” *Automation in Construction*, 87, 297–308.

Kim, K., and Kim, K. J. (2010). “Multi-agent-based simulation system for construction operations with congested flows.” *Automation in Construction*, Elsevier B.V., 19(7), 867–874.

Koskela, L., and Howell, G. (2001). “Reforming Project Management : the Role of Planning , Execution and Controlling.” *Proceedings of 9th International Group for Lean Construction Conference*, 185–198.

Ku, K., Tech, V., Mahabaleshwarkar, P. S., and Tech, V. (2011). “Bulding Interactive Modeling for Construction Education in Virtual Worlds.” *Journal of Information Technology in Construction (ITCon)*, 16(13), 189–208.

Lawson, R. M., Ogden, R. G., and Bergin, R. (2011). "Application of Modular Construction in High-Rise Buildings." *Journal of Architectural Engineering*, 18(2), 148–154.

Leu, S. Sen, Yang, C. H., and Huang, J. C. (2000). "Resource leveling in construction by genetic algorithm-based optimization and its decision support system application." *Automation in construction*, 10(1), 27–41.

Lewis, P., Leming, M., Frey, H. C., and Rasdorf, W. (2011). "Assessing Effects of Operational Efficiency on Pollutant Emissions of Nonroad Diesel Construction Equipment." *Journal of the Transportation Research Board*, 2233(1), 11–18.

Liang, D., and Thomaz, E. (2019). "Audio-Based Activities of Daily Living (ADL) Recognition with Large-Scale Acoustic Embeddings from Online Videos." *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 3(1), 1–18.

Liang, M., and Hu, X. (2015). "Recurrent convolutional neural network for object recognition." *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 07-12-June(Figure 1), 3367–3375.

Liu, H., Altaf, M. S., and Lu, M. (2015). "Automated production planning in panelized construction enabled by integrating discrete-event simulation and BIM Automation for heavy lift study View project Cloud-based Integrated Building Information Modelling View project." *5th International Construction Specialty COnference*, 1–10.

Louis, J., and Dunston, P. (2016a). "Platform for Real Time Operational Overview of

Construction Operations.” *Construction Research Congress ASCE*, 2039–2049.

Louis, J., and Dunston, P. S. (2016b). “Methodology for Real-Time Monitoring of Construction Operations Using Finite State Machines and Discrete-Event Operation Models.” *Journal of Construction Engineering and Management*, 143(3), 04016106.

Louis, J., Dunston, P. S., and Martinez, J. (2014). “Simulating and Visualizing Construction Operations using Robot Simulators and Discrete Event Simulation.” *International Conference on Computing in Civil and Building Engineering*, Orlando, Florida, 355–364.

Louis, J., and Martinez, J. (2012). “Rendering stereoscopic Augmented Reality scenes with occlusions using depth from stereo and texture mapping.” *Construction Research Congress 2012: Construction Challenges in a Flat World, Proceedings of the 2012 Construction Research Congress*, (May 2012), 1370–1380.

Lu, W., and Olofsson, T. (2014). “Building information modeling and discrete event simulation: Towards an integrated framework.” *Automation in Construction*, Elsevier B.V., 44, 73–83.

Luo, H., Xiong, C., Fang, W., Love, P. E. D., Zhang, B., and Ouyang, X. (2018). “Convolutional neural networks: Computer vision-based workforce activity assessment in construction.” *Automation in Construction*.

Lyon, R. F. (2010). “Machine Hearing: An Emerging Field.” *IEEE Signal Processing Magazine*, 27(5), 131–139.

Maccagno, A., Mastropietro, A., Mazziotta, U., Lee, Y., and Uncini, A. (2019). “A

CNN Approach for Audio Classification in Construction Sites." *The 29th Italian Workshop on Neural Networks*, Vietri sul Mare (SA), Italy, 1–10.

Mao, X., and Zhang, X. (2008). "Construction Process Reengineering by Integrating Lean Principles and Computer Simulation Techniques." *Journal of Construction Engineering and Management*, 134(5), 371–381.

Martín-Garín, A., Millán-García, J. A., Baïri, A., Millán-Medel, J., and Sala-Lizarraga, J. M. (2018). "Environmental monitoring system based on an Open Source Platform and the Internet of Things for a building energy retrofit." *Automation in Construction*, Elsevier, 87(March 2017), 201–214.

Mathur, N., Aria, S. S., Adams, T., Ahn, C. R., and Lee, S. (2015). "Automated Cycle Time Measurement and Analysis of Excavator's Loading Operation Using Smart Phone-Embedded IMU Sensors." *International Workshop on Computing in Civil Engineering*, ASCE, Austin, TX, 215–222.

Mirdamadi, S., Fontanili, F., and Dupont, L. (2007). "Discrete Event Simulation-Based Real-Time Shop Floor Control." *21st European Conference on Modelling and Simulation*, Prague, Czech Republic.

Moghadam, M. (2014). "Lean-Mod: An Approach to Modular Construction Manufacturing Production Efficiency Improvement." University of Alberta.

Murray, N., Fernando, T., and Aouad, G. (2003). "A Virtual Environment for the Design and Simulated Construction of Prefabricated Buildings." *Virtual Reality*, 6(4), 244–256.

Nahmens, I., and Ikuma, L. H. (2012). "Effects of lean construction on sustainability of modular homebuilding." *Journal of Architectural Engineering*, 18(2), 155–163.

Naoyuki Kanda, R. T. and Y. O. (2013). "ELASTIC SPECTRAL DISTORTION FOR LOW RESOURCE Research Laboratory , Hitachi Ltd ." 309–314.

Nath, N. D., Chaspari, T., and Behzadan, A. H. (2018). "Automated ergonomic risk monitoring using body-mounted sensors and machine learning." *Advanced Engineering Informatics*, 38(2018), 514–526.

Neill, G. O., Ball, M., Liu, Y., Noghabaei, M., and Han, K. (2020). "Toward Automated Virtual Assembly for Prefabricated Construction: Construction Sequencing through Simulated BIM." *Construction Research Congress*, Reston, VA, 802–811.

Ng, S. T., Shi, J., and Fang, Y. (2009). "Enhancing the logistics of construction materials through activity-based simulation approach." *Engineering, Construction and Architectural Management*, 16(3), 224–237.

Ok, S. C., and Sinha, S. K. (2006). "Construction equipment productivity estimation using artificial neural network model." *Construction Management and Economics*.

Ordóñez, F. J., and Roggen, D. (2016). "Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition." *Sensors (Switzerland)*, 16(1).

Park, T., Kang, T., Lee, Y., and Seo, K. (2014). "Project Cost Estimation of National Road in Preliminary Feasibility Stage Using BIM/GIS Platform." *International*

Conference in Computing in Civil and Building Engineering, ASCE, Orlando, Florida, 423–430.

Ponz-Tienda, J. L., Yepes, V., Pellicer, E., and Moreno-Flores, J. (2013). “The Resource Leveling Problem with multiple resources using an adaptive genetic algorithm.” *Automation in Construction*, Elsevier B.V., 29, 161–172.

Radford, A., Metz, L., and Chintala, S. (2015). “Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks.” 1–16.

Rashid, K. M., and Behzadan, A. H. (2018). “Risk Behavior-Based Trajectory Prediction for Construction Site Safety Monitoring.” *Journal of Construction Engineering and Management*, 144(2), 04017106.

Rashid, K. M., Datta, S., and Behzadan, A. H. (2017). “Coupling risk attitude and motion data mining in a preemtive construction safety framework.” *Proceeding of Winter Simulation Conference*, IEEE, Las Vegas, NV, 4220–4227.

Rashid, K. M., and Louis, J. (2019a). “Construction Equipment Activity Recognition from IMUs Mounted on Articulated Implements and Supervised Classification.” *ASCE International Conference on Computing in Civil Engineering*, Atlanta, Georgia, 130–138.

Rashid, K. M., and Louis, J. (2019b). “Times-series data augmentation and deep learning for construction equipment activity recognition.” *Advanced Engineering Informatics*, 42(April), 100944.

Rashid, K. M., and Louis, J. (2019c). “Time-Warping: A Time Series Data Augmentation of IMU Data for Construction Equipment Activity Identification.”

Proceedings of the 36th International Symposium on Automation and Robotics in Construction (ISARC), 651–657.

Rashid, K. M., and Louis, J. (2020). "Automated Activity Identification for Construction Equipment Using Motion Data From Articulated Members." *Frontiers in Built Environment*, 5(January), 1–14.

Rezazadeh Azar, E., and McCabe, B. (2012). "Part based model and spatial-temporal reasoning to recognize hydraulic excavators in construction images and videos." *Automation in Construction*, 24, 194–202.

Richard, G., Sundaram, S., and Narayanan, S. (2013). "An Overview on Perceptually Motivated Audio Indexing and Classification." *Proceedings of the IEEE*, 101(9), 1939–1954.

Ruwanpura, J. Y., and Ariaratnam, S. T. (2007). "Simulation modeling techniques for underground infrastructure construction processes." *Tunnelling and Underground Space Technology*, 22(5–6), 553–567.

Sabillon, C., Rashidi, A., Samanta, B., Davenport, M. A., and Anderson, D. V. (2020). "Audio-Based Bayesian Model for Productivity Estimation of Cyclic Construction Activities." *Journal of Computing in Civil Engineering*, 34(1), 1–14.

Schindler, A., Lidy, T., and Rauber, A. (2016). "Comparing Shallow versus Deep Neural Network Architectures for Automatic Music Genre Classification." *9th Forum Media Technology*, St. Poelten, Austria, 17–21.

Schlüter, J., and Grill, T. (2013). "Exploring Data Augmentation for Improved

Singing Voice Detection With Neural Networks."

Seo, J., Han, S., Lee, S., and Kim, H. (2015). "Computer vision techniques for construction safety and health monitoring." *Advanced Engineering Informatics*.

Shehata, M. E., and El-Gohary, K. M. (2011). "Towards improving construction labor productivity and projects' performance." *Alexandria Engineering Journal*, Faculty of Engineering, Alexandria University, 50(4), 321–330.

Sherafat, B., Rashidi, A., Lee, Y.-C., and Ahn, C. R. (2019). "Automated Activity Recognition of Construction Equipment Using a Data Fusion Approach." *ASCE International Conference on Computing in Civil Engineering*, Atlanta, Georgia, 1–8.

Shitole, V., Louis, J., and Tadepalli, P. (2019). "Optimizing Earth Moving Operations Via Reinforcement Learning." *Proceedings - Winter Simulation Conference*, 2019-Decem, 2954–2965.

Sokovic, M., Pavletic, D., and Pipan, K. K. (2010). "Quality Improvement Methodologies - PDCA Cycle, RADAR Matrix, DMAIC and DFSS." *Journal of achievements in materials and manufacturing engineering*, 43(1), 476–483.

Song, L., and Eldin, N. N. (2012). "Adaptive real-time tracking and simulation of heavy construction operations for look-ahead scheduling." *Automation in Construction*, 27, 32–39.

Spriggs, E. H., De La Torre, F., and Hebert, M. (2009). "Temporal segmentation and activity classification from first-person sensing." *2009 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2009*, 17–24.

Su, X., Tong, H., and Ji, P. (2014). "Activity recognition with smartphone sensors." *Tsinghua Science and Technology*, 19(3), 235–249.

Su, Y. Y., and Liu, L. Y. (2007). "Real-time Construction Operation Tracking from Resource Positions." *Computing in Civil Engineering*, Intergovernmental Panel on Climate Change, ed., Cambridge University Press, Cambridge, 1–30.

Teicholz, P. (2013). "Labor-Productivity Declines in the Construction Industry: Causes and Remedies (a second look)." *AECBytes*.

Teizer, J., Venugopal, M., and Walia, A. (2018). "Ultrawideband for Automated Real-Time Three-Dimensional Location Sensing for Workforce, Equipment, and Material Positioning and Tracking." *Transportation Research Record: Journal of the Transportation Research Board*, 56–64.

Tremblay, S., Fortin-Simard, D., Blackburn-Verreault, E., Gaboury, S., Bouchard, B., and Bouzouane, A. (2015). *Exploiting environmental sounds for activity recognition in smart homes*. *AAAI Workshop: Artificial Intelligence Applied to Assistive Technologies and Smart Environments*.

Um, T. T., Pfister, F. M. J., Pichler, D., Endo, S., Lang, M., Hirche, S., Fietzek, U., and Kulić, D. (2017). "Data Augmentation of Wearable Sensor Data for Parkinson's Disease Monitoring using Convolutional Neural Networks."

Urbanowicz, R. J., Meeker, M., La Cava, W., Olson, R. S., and Moore, J. H. (2018). "Relief-based feature selection: Introduction and review." *Journal of Biomedical Informatics*, 85, 189–203.

Vahdatikhaki, F., and Hammad, A. (2014). "Framework for near real-time simulation of earthmoving projects using location tracking technologies." *Automation in Construction*, 42, 50–67.

Vahdatikhaki, F., Hammad, A., and Setayeshgar, S. (2013). "Real-time simulation of earthmoving projects using automated machine guidance." *Proceedings of the International Symposium on Automation and Robotics in Construction*, Montreal, Canada, 1.

Vepakomma, P., De, D., Das, S. K., and Bhansali, S. (2015). "A-Wristocracy: Deep learning on wrist-worn sensing for recognition of user complex activities." *2015 IEEE 12th International Conference on Wearable and Implantable Body Sensor Networks, BSN 2015*, IEEE, 1–6.

Walse, K., and Dharaskar, R. V. (2016). "PCA-based optimal ANN classifiers for human activity recognition using mobile sensors data." 50(January).

Wang, P., and Abourizk, S. M. (2009). "Large-scale simulation modeling system for industrial construction." *Canadian Journal of Civil Engineering*, 36(9), 1517–1529.

Wang, X., Love, P. E. D., and Davis, P. R. (2012). "RFID+ for Tracking Earthmoving Operations." *Construction Research Congress*, (Mathiassen 1993), 778–786.

Yan, X., Li, H., Wang, C., Seo, J. O., Zhang, H., and Wang, H. (2017). "Development of ergonomic posture recognition technique based on 2D ordinary camera for construction hazard prevention through view-invariant features in 2D skeleton motion." *Advanced Engineering Informatics*.

Yang, J. B., Nguyen, M. N., San, P. P., Li, X. L., and Krishnaswamy, S. (2015). "Deep convolutional neural networks on multichannel time series for human activity recognition." *IJCAI International Joint Conference on Artificial Intelligence*, 2015-Janua(Ijcai), 3995–4001.

Yang, Z., Ma, Z., and Wu, S. (2016). "Optimized flowshop scheduling of multiple production lines for precast production." *Automation in Construction*, The Authors, 72, 321–329.

You, S., Kim, J.-H., Lee, S., Kamat, V., and Robert, L. P. (2018). "Enhancing perceived safety in human–robot collaborative construction using immersive virtual environments." *Automation in Construction*, Elsevier, 96(March), 161–170.

Yu, H., Al-Hussein, M., Al-Jibouri, S., and Telyas, A. (2013). "Lean transformation in a modular building company: A case for implementation." *Journal of Management in Engineering*, 29(1), 103–111.

Zeng, M., Nguyen, L. T., Yu, B., Mengshoel, O. J., Zhu, J., Wu, P., and Zhang, J. (2014). "Convolutional Neural Networks for Human Activity Recognition using Mobile Sensors." *Proceedings of the 6th International Conference on Mobile Computing, Applications and Services*, 6.

Zhai, D., Goodrum, P. M., Haas, C. T., and Caldas, C. H. (2009). "Relationship between Automation and Integration of Construction Information Systems and Labor Productivity." *Journal of Construction Engineering and Management*, 135(8), 746–753.

Zhang, H., Zhai, D., and Yang, Y. N. (2014). “Simulation-based estimation of environmental pollutions from construction processes.” *Journal of Cleaner Production*, Elsevier Ltd, 76, 85–94.

Zhang, X. (2004). “Production Process for Manufactured Housing.” *Journal of Construction Engineering and Management*, 130(2), 235–244.

Zou, J., and Kim, H. (2007). “Using Hue, Saturation, and Value Color Space for Hydraulic Excavator Idle Time Analysis.” *Journal of Computing in Civil Engineering*, 21(4), 238–246.